

Automatic Extrinsic Calibration Method for LiDAR and Camera Sensor Setups

Jorge Beltrán, Carlos Guindel, and Fernando García, *Member, IEEE*

Abstract—Most sensor setups for onboard autonomous perception are composed of LiDARs and vision systems, as they provide complementary information that improves the reliability of the different algorithms necessary to obtain a robust scene understanding. However, the effective use of information from different sources requires an accurate calibration between the sensors involved, which usually implies a tedious and burdensome process. We present a method to calibrate the extrinsic parameters of any pair of sensors involving LiDARs, monocular or stereo cameras, of the same or different modalities. The procedure is composed of two stages: first, reference points belonging to a custom calibration target are extracted from the data provided by the sensors to be calibrated, and second, the optimal rigid transformation is found through the registration of both point sets. The proposed approach can handle devices with very different resolutions and poses, as usually found in vehicle setups. In order to assess the performance of the proposed method, a novel evaluation suite built on top of a popular simulation framework is introduced. Experiments on the synthetic environment show that our calibration algorithm significantly outperforms existing methods, whereas real data tests corroborate the results obtained in the evaluation suite. Open-source code is available at https://github.com/beltransen/velo2cam_calibration.

Index Terms—Automatic calibration, extrinsic parameters, LiDAR, monocular cameras, stereo cameras

I. INTRODUCTION

AUTONOMOUS driving relies on accurate information about the environment to make proper decisions concerning the trajectory of the vehicle. High-level inference modules receive these data from the perception systems, which must be therefore endowed with an exceptional degree of robustness under different circumstances such as illumination and weather.

Consequently, the design of perception systems intended for onboard automotive applications is currently geared towards topologies with several complementary sensory modalities. Vision systems are frequent in close-to-market vehicle setups [1] due to their ease of integration and their ability to provide appearance information. Stereo-vision systems, which use a pair of cameras separated a fixed distance to get depth information about the environment, stand out as a cost-effective solution able to provide additional dense 3D information to model the surroundings of the vehicle.

This work was funded by the Spanish Government (PID2019-104793RB-C31 and RTI2018-096036-B-C21), the Universidad Carlos III of Madrid (PEVAUTO-CM-UC3M), and the Comunidad de Madrid (SEGVAUTO-4.0-CM P2018/EMT-436).

The authors are with the Intelligent Systems Lab (LSI), Universidad Carlos III de Madrid, Leganés, 28911 Spain e-mail: {jbeltran, cguindel, fegarcia}@ing.uc3m.es).

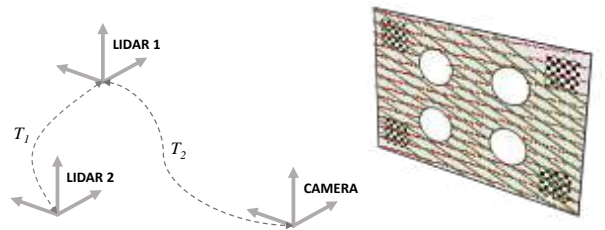


Fig. 1. Sample calibration scenario for an arbitrary setup with a camera and two LiDAR scanners, where the calibration target is placed in the overlapping field of view of the involved sensors.

On the other hand, the remarkable development of 3D laser scanning technology has enabled its widespread use in both research and industry driving applications in recent years. Unlike vision systems, LiDAR range measurements are accurate and, frequently, provide information in a full 360° field of view. Setups made of more than one LiDAR device are becoming more and more popular since they allow gathering high-resolution data using compact setups.

Due to the particular features of these sensory technologies, they are suitable to be part of the same perception system, providing complementary information. In that kind of design, data from the different sensors must be appropriately combined before inference making use of fusion techniques [2], [3]. In the most usual setup, sensors have overlapping fields of view (as in Fig. 1), and the advantages conferred by their joint use come from the ability to make correspondences between both data representations. This is the case, for example, with popular multi-modal 3D object detection approaches such as F-PointNet [4] or AVOD [5]. These methods assume that an accurate estimate of the relative pose between the sensors, given by their extrinsic parameters, has been obtained beforehand through a calibration process and is available during operation.

However, multi-modal calibration is a problem that is still far from trivial. Existing calibration methods suffer from different problems, such as the need for burdensome ad-hoc environments or the lack of applicability to custom sensor setups. It is noteworthy that automotive setups require extraordinary accuracy in the calibration so that it is still valid for data association at long distances.

In this work, we present a calibration method tailored to automotive sensor setups composed of vision devices and multi-layer LiDAR scanners. Unlike the existing methods, no strong assumptions are made, allowing its use with medium-resolution scanners (e.g., 16-layer devices) and a wide range of poses between sensors. As a continuation of the work

presented in [6], we propose a method suitable to handle different combinations of sensor pairs, including monocular vision devices. Our method can be performed within a reasonable time using a simple setup designed to exploit the correspondences in the data from both devices.

The remainder of this paper is organized as follows. In Section II, a brief review of related work is provided. Section III is employed to present a general overview of the proposed algorithm. In Sections IV and V, the details of the different stages of the approach are described. Section VI provides experimental results that assess the performance of the method. Finally, conclusions and open issues are discussed in Section VII.

II. RELATED WORK

The issue of calibration of extrinsic parameters expressing the relative pose of sensors of different modalities has been addressed by many researchers in the past, driven by its frequent application in robotics and automotive platforms. The camera-to-range problem has attracted considerable attention, although multi-camera and, more recently, multi-LiDAR systems have also been a subject of interest in the literature.

Calibration is frequently assumed as a process to be performed in a controlled environment before the regular operation of the perception stack. Traditional methods require manual annotation to some extent [7]. However, since miscalibrations are common in robotic platforms, research effort has usually focused on automatic approaches. As the process aims to find the correspondence between data acquired from different points of view, unambiguous fiducial instruments have been used as calibration targets, such as triangular boards [8], polygonal boards [9], spheres [10], and boxes [11]. Such diversity of shapes deals with the necessity of the targets to be distinguishable in all data representations from sensors. Nonetheless, planar targets are particularly prevalent [12] since they are easily detectable using range information and provide a characteristic shape that can be used to perform geometrical calculations. When monocular cameras are involved, the addition of visual features into the target, such as checkerboards [13] or QR markers [14], allows retrieving the geometry of the scene by inferring the missing scale factor.

With the widespread introduction of LiDAR sensors providing high-resolution 3D point clouds in recent years, research interest has shifted to devices of this kind. Geiger et al. [15] proposed a calibration method based on a single shot in the presence of a setup based on several planar checkerboards used as calibration targets. Velas et al. [16] proposed an approach enabling the estimation of the extrinsic parameters using a single point of view, based on the detection of circular features on a calibration pattern. Similarly, Zhou et al. [17] made use of a checkerboard to solve the calibration problem by finding correspondences between its representations in LiDAR and image data, using either one or several poses. In general, these methods are targeted to dense range measurements so that 3D LiDAR scanners with lower resolution (e.g., the 16-layer scanner used in this work) entail particular issues that are addressed in this paper. Due to the popularity of this modality,

some works are also being devoted to the topic of extrinsic calibration between multiple LiDAR scanners [18].

A relevant second group of approaches dispenses with any artificial calibration targets and use the features in the environment. Moghadam et al. [19] use linear features extracted from natural scenes to determine the transformation between the coordinate frames. Usually, these methods are suitable for indoor scenes populated with numerous linear landmarks, although some recent works have made efforts to adapt them to outdoor applications [20]. In traffic environments, the ground plane and the obstacles have been used to perform camera-laser calibration [21], although some parameters are assumed as known. Other approaches are based on semi-automatic methods [22] that perform registration on user-selected regions. More recently, Schneider et al. [23] took advantage of a deep convolutional neural network to perform all the calibration steps in a continuous online procedure. CalibNet [24] has been proposed as a self-supervised calibration framework where the network is trained to minimize the geometric and photometric errors. However, models of this type are difficult to apply to custom sensor setups as they require a prior training stage.

On the other hand, the assessment of calibration methods remains an open issue, given that an accurate ground-truth of the six parameters defining the relationship between the pose of the sensors cannot be obtained in practice. The lack of standard evaluation metrics has led to the use of custom schemes, which are difficult to extend to other domains and eventually based on inaccurate manual annotations. In this regard, Levinson and Thrun [25] presented a method to detect miscalibrations through the variations in an objective function computed from the discontinuities in the scene. A different approach was proposed by Pandey et al. [26], who performed calibration through the maximization of mutual information computed using LiDAR reflectivity measurements and camera intensities.

Nevertheless, current calibration methods still do not provide a comprehensive response to the need to estimate the extrinsic parameters of certain sensor setups, such as the ones found in autonomous driving. They are either excessively focused on specific configurations, lacking generalization ability, or have not been sufficiently validated due to the unavailability of objective assessment methods. We intend to provide a universal approach able to perform calibration in a wide variety of setups and situations, including those that are usually overlooked, and prove its adequacy quantitatively through a novel benchmark that allows fair comparison with existing methods.

III. METHOD OVERVIEW

We present a method to estimate the rigid-body transformation that defines the relative pose between a pair of sensors. Each of these sensors can be a LiDAR scanner, a monocular camera, or a stereo camera, in any possible combination.

The transformation between the pair of sensors can be defined by a vector of six parameters $\theta = (t_x, t_y, t_z, r_x, r_y, r_z)$, which describe the position and rotation of one of the devices in the reference frame attached to the other one. Rotations

around the axes (r_x, r_y, r_z) are usually referred to as roll, pitch, and yaw angles.

Parameters in θ unambiguously define a matrix \mathbf{T} that can be used to transform a 3D point between the two coordinate systems. For instance, in a LiDAR-monocular setup, a point \mathbf{p}_M in monocular coordinates, $\{M\}$, can be transformed into LiDAR space, $\{L\}$, by means of $\mathbf{p}_L = \mathbf{T}_{LM}\mathbf{p}_M$ once the transformation matrix \mathbf{T}_{LM} is built. Note that, in that particular case, the parameters θ_{LM} , used to obtain \mathbf{T}_{LM} , express the pose of $\{M\}$ with respect to $\{L\}$.

With the proposed approach, the transformation is obtained automatically from data retrieved by the sensors to be calibrated. A custom-made planar target is used to provide features that are detected and paired between both data representations. As noticeable in the two different embodiments shown in Fig. 2, this calibration pattern is endowed with geometrical and visual characteristics that enable the estimation of keypoints in LiDAR, stereo, and monocular modalities. On the one hand, four circular holes are used to take advantage of geometrical discontinuities in LiDAR and stereo point clouds. On the other hand, four ArUco markers [27] are placed close to the target corners so that 3D information can be inferred from monocular images.



Fig. 2. Two different embodiments of the custom calibration pattern

The method does not impose severe limits on the relative pose between the devices and is therefore suitable for sensor setups where the magnitudes of the translation and rotation parameters are substantial. Only two reasonable constraints are required. First of all, there has to be an overlapping area between the sensors' field of view, where the calibration target is to be placed. Secondly, the holes in the pattern must be well visible in the data retrieved by the sensors; in particular, whenever range data is involved in the calibration, each circle must be represented by at least three points. In the case of multi-layer LiDAR sensors, this means that at least two scan planes intersect with each of the circles. Moreover, the parameters intrinsic to each device (e.g., focal lengths or stereo baseline) are assumed known.

The procedure is designed to be performed in a static environment. Although the method can provide a quick estimate of the extrinsic parameters with just one pose of the calibration target, it is possible to increase the accuracy and robustness of the results by accumulating several positions, as will be shown later.

The proposed calibration algorithm, illustrated in Fig. 3, is divided into two different stages: the first one involves the segmentation of the calibration target and the localization of

the reference points in each of the sensors' coordinate systems; on the other hand, the second one performs the computation of the transformation parameters that enable the registration of the reference points.

IV. TARGET SEGMENTATION

This first stage aims to localize the calibration target in each sensor's data. Consequently, the measurements at this stage are relative to the local coordinate system of the corresponding sensor. As the features used to localize the pattern are different for each modality, three different variants of the procedure are proposed here, one per sensor type. In all cases, the output of this stage is a set of four 3D points representing the center of the holes in the target, in local coordinates. These points will be later used to find correspondences between the different data sources.

Although the processing of LiDAR and stereo data has some differences, especially at the beginning of the segmentation stage, both share a common trunk once the useful range data is represented in a 3D point cloud structure. The monocular alternative is substantially different as it relies on the ArUco markers instead.

The procedure described in this section is intended to be applied to every data frame provided by the corresponding sensor. Data from all sensors are processed in parallel, so they do not have to share a common trigger nor have identical refresh rates, as long as the scene is static.

A. LiDAR Data Preprocessing

Data from a LiDAR scanner is assumed to be represented as a 3D point cloud, \mathcal{P}_0^L , with measurements distributed into different layers, as typical in mechanical devices based on rotating mirrors. Before feeding the data to the segmentation procedure, pass-through filters are applied in the three cartesian coordinates to remove points outside the area where the target is to be placed, avoiding spurious detections that could slow down the processing. The limits of the pass-through filters must be set according to the location and size of the sensors' overlapping area. The resulting cloud, \mathcal{P}_1^L , must represent both the calibration target and the points behind it visible from the LiDAR through the holes.

As a first step towards segmenting the holes in the pattern, the points representing the edges of the target must be extracted. For the LiDAR modality, we follow the method in [25] to find depth discontinuities. Each point in the cloud, $\mathbf{p}_i \in \mathcal{P}_1^L$, is assigned a magnitude representing the depth gradient with respect to their neighbors:

$$p_{i,\Delta} = \max(p_{i-1,r} - p_{i,r}, p_{i+1,r} - p_{i,r}, 0) \quad (1)$$

Where $p_{i,r}$ is the range measurement given by the sensor for the point \mathbf{p}_i (i.e., the spherical radius coordinate), and \mathbf{p}_{i-1} and \mathbf{p}_{i+1} are the points adjacent to \mathbf{p}_i in the same scan plane. Then, we filter out all points \mathbf{p}_i with a discontinuity value $p_{i,\Delta} < \delta_{\text{discont},L}$, resulting in \mathcal{P}_2^L . Note that this procedure assumes that measures from rays passing through the holes exist, so they must collide with some solid located behind the target within the measurement range of the LiDAR.

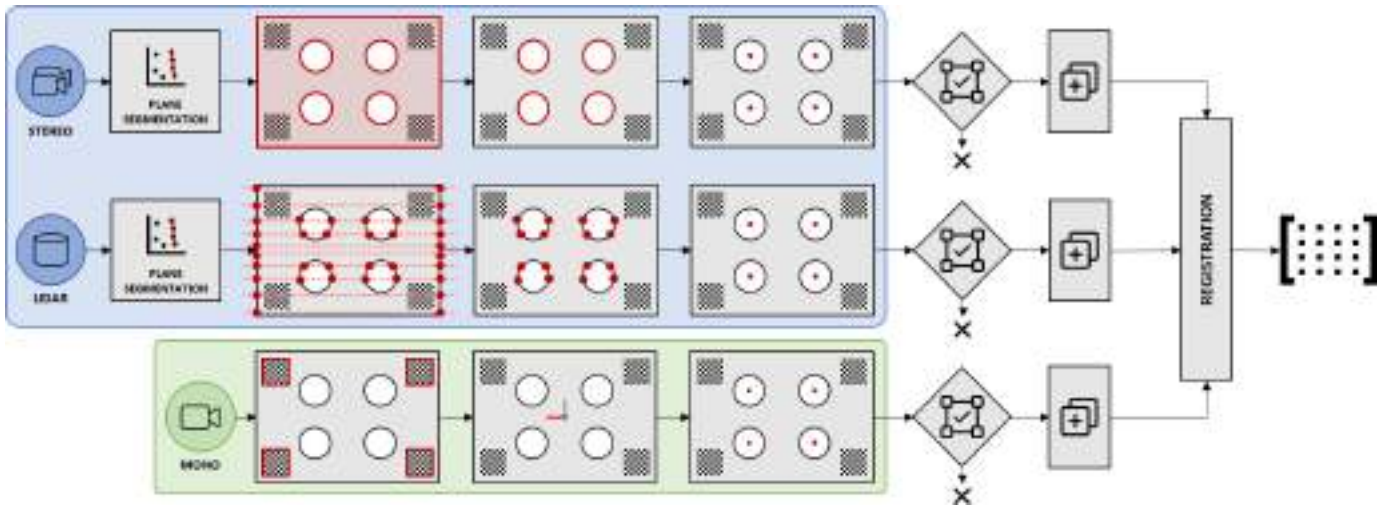


Fig. 3. Overview of the different stages of the presented method. For 3D inputs (in blue): plane segmentation, target detection, circles segmentation, and reference points estimation. For monocular cameras (in green): ArUco markers detection, estimation of the target's 3D pose, and reference points estimation. Then, for each frame and modality: geometric consistency check, point aggregation, and sensor registration.

B. Stereo Data Preprocessing

When one of the sensors to be calibrated is a stereo-vision system, data processing starts by converting the raw image pair into a 3D point cloud using a stereo matching procedure. In our experiments, we use the Semi-Global Block Matching (SGBM) variant of [28] implemented by OpenCV, which we found reasonably accurate for depth estimation. Note that, when this modality is involved, the calibration target is expected to have some texture (e.g., wood grain) so that the stereo correspondence problem can be successfully solved. However, in our experiments, we found that the intensity differences caused by the pattern borders themselves are generally sufficient. Since the system is assumed canonical and the baseline between cameras known, points can be then provided with an estimate of their depth, and a 3D point cloud \mathcal{P}_0^S can be straightforwardly obtained using the pinhole model.

Similar to the LiDAR branch, pass-through filters are applied to \mathcal{P}_0^S to limit the search space. However, for the stereo modality, the extraction of the points representing the target edges in the filtered cloud, \mathcal{P}_1^S , relies on the appearance information provided by one of the images of the stereo pair. Concretely, a Sobel filter is applied over the image, and then, all points in \mathcal{P}_1^S that map to pixels with a low value in the Sobel image (smaller than $\tau_{\text{sobel},S}$) are filtered out, producing \mathcal{P}_2^S . In this way, edge segmentation is less affected by inaccuracies in border localization, which are frequent in stereo matching.

C. Range Data

The steps followed to segment the pattern holes in the preprocessed point clouds are common for both the LiDAR and stereo modalities. The intended outcome is an estimate of the 3D location of the centers in sensor coordinates.

1) *Plane Segmentation*: First of all, a plane segmentation using RANSAC is applied to \mathcal{P}_1 (the cloud resulting from the pass-through filters, either \mathcal{P}_1^L or \mathcal{P}_1^S), which provides a

plane model π representing the calibration target. To ensure the model's accuracy, we use a tight RANSAC threshold δ_{plane} and impose that the plane must be roughly vertical in sensor coordinates, with a tolerance α_{plane} . If it is impossible to find a plane that fits the data, the current frame is discarded.

Afterward, the plane model π is employed in \mathcal{P}_2 (i.e., the cloud representing the edges of the pattern) to remove all the points not belonging to the plane. A threshold of δ_{inliers} is considered for the inliers. Consequently, the new cloud \mathcal{P}_3 contains only points representing the edges of the calibration target; that is, the outer borders and the holes.

2) *Transformation to 2D Space*: As all the remaining points belong to the same plane, dimensionality reduction is performed at this point. This is implemented by transforming \mathcal{P}_3 so that the XY-plane coincides with π and projecting all the 3D points onto π . Points in the resulting \mathcal{P}_4 cloud are, therefore, in 2D space.

3) *Circle Segmentation*: Next, 2D circle segmentation is used to extract a model of the pattern holes present in \mathcal{P}_4 . This step is performed iteratively in a process that seeks out the most supported circle and removes its inliers before starting the search for the next one. Iterations continue until the remaining points are not enough to describe a circle. If at least four circles have been found, the procedure moves forward; otherwise, the current frame is not considered. Inliers are required to be below a threshold of δ_{circle} from the model, and only circles within a radius tolerance of δ_{radius} are considered.

The points found in the circle segmentation procedure are checked for geometric consistency with the dimensions of the pattern. To that end, the centers are grouped in sets of four, and the dimensions of the rectangle that they form (diagonal, height, width, and perimeter) are compared with the theoretical ones, with a tolerance $\delta_{\text{consistency}}$ expressed as a percentage of deviation from the expected values. Presumably, only one set of centers will fulfill these restrictions; if either none or more than one sets pass the check, the frame is discarded. This step is intended to prune out spurious detections that may occur

due to confusion with other elements in the scene.

Once the holes are correctly identified, their centers are converted back from the 2D space defined by π to the 3D space in sensor coordinates, forming the cloud \mathcal{P}_p . Note that \mathcal{P}_p must contain exactly four points.

D. Monocular Data

If the sensor to be calibrated is a monocular camera, the extraction of the reference points requires the detection of ArUco markers, which provide the cues necessary to retrieve the geometry of the target.

ArUco markers are synthetic square markers made of a black border and an inner binary matrix designed to allow its unequivocal identification [27]. In our calibration target, four ArUco markers are used, one on each corner; due to this location, they do not affect either target or hole detection by other modalities.

As both the camera's intrinsic parameters and the marker dimensions are known, it is possible to retrieve the 3D pose of each marker with respect to the camera through the resolution of a classic perspective-n-point (PnP) problem. In our implementation, we handle our four-marker setup as an *ArUco board*, which allows estimating the pose of the calibration target accurately by using all the markers jointly. An iterative Levenberg-Marquardt optimization is carried out to find the board pose that minimizes the reprojection error [29], using the average pose of the four individual markers as an initial guess. As a result, the 3D position of the center of the board is obtained, along with its orientation in space.

To generate a set of four points equivalent to the \mathcal{P}_p clouds obtained from range data, we extract the points representing the center of the reference holes by taking advantage of the fact that their relative positions in the calibration target are known. These points constitute the resulting four-point cloud \mathcal{P}_p^M .

E. Point Aggregation and Clustering

At the end of the segmentation stage, two clouds \mathcal{P}_p must have been generated, one per sensor involved in the calibration. Each represents the 3D location of the reference points (the centers of the target holes) for a single static scene in the coordinate frame attached to the respective sensor.

These data would be enough to find the transform representing the relative pose of the sensors. However, different sources of noise inherent to the method (e.g., sensor noise and non-deterministic procedures such as RANSAC) can affect the accuracy of the result. To increase the robustness of the algorithm, we augment the information available by repeatedly applying the segmentation step and accumulating the results in two different ways.

1) *Accumulation over Several Data Frames*: Since it is usually feasible to maintain the calibration scene static for a certain period, we accumulate the points that compose \mathcal{P}_p over N data frames to generate \mathcal{P}_p' and then perform Euclidean clustering on this cumulative cloud. If more than four clusters are found, data is considered unreliable and not used for registration; otherwise, cluster centroids, stored in the

resulting cloud \mathcal{P}_c , are employed as a consolidated estimate of the centers' locations. The clustering parameters, namely cluster tolerance δ_{cluster} , minimum cluster size $N_{\text{cluster,min}}$, and maximum cluster size $N_{\text{cluster,max}}$, depend on the number of iterations taken into account.

According to the experimental results shown later, we usually adopt $N = 30$, which offers satisfactory results in a limited timeframe. Naturally, the time necessary to complete the procedure depends on the sensor's framerate but is rarely longer than a few seconds.

2) *Accumulation over Several Target Poses*: As will be shown later, the method can deliver an estimated calibration with a single target position. However, it is possible to increase the accuracy of the estimation by considering more than four reference points. If the segmentation procedure is repeated for M different poses of the calibration target with respect to the sensors, the \mathcal{P}_c clouds obtained with each pose are accumulated in a \mathcal{P}_c' cloud where $4 \times M$ reference points are available to perform the registration stage.

If the poses of the target are selected so that the resulting reference points are not coplanar and cover a wide range of distances from the sensors, the additional constraints provided by the new poses solve possible ambiguities and improve the overall quality of the final calibration.

V. REGISTRATION

As a result of the segmentation stage, two clouds \mathcal{P}_c' , one per sensor, are obtained. They contain the estimated 3D location of the centers of the circles expressed in sensor coordinates; that is, with respect to a frame attached to the sensor.

The goal of the registration step is to find the optimal transformation parameters $\hat{\theta}$ so that when the resulting transformation \hat{T} is applied, it results in the best alignment (i.e., minimum distance) between the reference points obtained from both sensors. Note that the approach has been designed to handle only two sources at a time so that the problem can be viewed as a multi-objective optimization with $4 \times M$ objective functions.

Before that, the registration procedure needs that each point in one of the \mathcal{P}_c' clouds is correctly paired with its homologous in the other cloud; that is, pairs of points representing the same reference points in both clouds must be associated.

A. Point Association

A point association procedure has been developed to avoid assuming that reference points in both \mathcal{P}_c clouds have the same ordering in their respective coordinate frames. Note that this condition would not be fulfilled when calibrating a front-facing 360° LiDAR scanner and a rear-looking camera, for instance.

Therefore, we convert the four centers in each \mathcal{P}_c to spherical coordinates and only assume that the point that appears highest in the cloud, that is, the one with the lowest inclination angle, belongs to the upper row of the calibration target (i.e., either the top-left or the top-right circle).

Distances from this point to the other three determine the correct ordering. In that way, each point can be associated

with the circle in the calibration target that it represents: top-left (tl), top-right (tr), bottom-left (bl), and bottom-right (br). The procedure is repeated for each of the M poses of the calibration target, so that each point \mathbf{p}_i in \mathcal{P}'_c is provided with labels $p_{i,a}$ and $p_{i,m}$ containing the hole in the pattern and the pose to which it corresponds, respectively:

$$p_{i,a} \in \{tl, tr, bl, br\} \quad (2)$$

$$p_{i,m} \in \{0, \dots, M\} \quad (3)$$

B. Solution

Later, the two resulting clouds, obtained from two arbitrary modalities X and Y and denoted here by \mathcal{P}'_c^X and \mathcal{P}'_c^Y , undergo a Umeyama registration procedure [30], responsible for finding the rigid transformation that minimizes the distance between their corresponding points. That is, assuming that the points in each cloud, $\mathbf{p}_i^X \in \mathcal{P}'_c^X$ and $\mathbf{p}_i^Y \in \mathcal{P}'_c^Y$, are ordered so that, $\forall i$:

$$p_{i,a}^X = p_{i,a}^Y \wedge p_{i,m}^X = p_{i,m}^Y \quad (4)$$

Then, the desired transformation matrix $\hat{\mathbf{T}}_{XY}$ is the one that minimizes the least-squares error criterion given by:

$$\frac{1}{4 \cdot M} \sum_{i=0}^{4 \cdot M} \|\mathbf{p}_i^X - \mathbf{T}_{XY} \mathbf{p}_i^Y\|^2 \quad (5)$$

This optimization problem is solved through singular value decomposition (SVD) and provides a closed-form solution from which the set of parameters expressing the relative position between both sensors, $\hat{\boldsymbol{\theta}}_{XY}$, can be straightforwardly retrieved. Conveniently, the Umeyama method handles singular situations where all the points are coplanar, as is the case when a single pattern position ($M = 1$) is used, thus avoiding misjudging them as reflections.

VI. EXPERIMENTS

The validation of the proposed approach has been addressed from two different perspectives. On the one hand, tests on a realistic synthetic test suite have been performed to retrieve plentiful quantitative data with respect to perfect ground-truth. On the other hand, the method has also been applied in a real environment to prove the validity of the approach in real use cases.

All the experiments were carried out without user intervention, except for the tuning of the pass-through filters mentioned in Sec. IV, which must be coarsely adapted to the location of the calibration pattern. The rest of the parameters were set to a fixed value for all the experiments, as reported in Table I. Unless otherwise stated, reference points are accumulated over 30 frames ($N = 30$); however, it should be noted that every frame delivered by the sensors counts toward this limit, regardless of whether a four-point solution has been extracted from it. Conversely, only successful frames (N') are taken into account for the cluster size limits.

TABLE I
SETTING OF CONSTANT PARAMETERS IN THE METHOD

Parameter	Description
Preprocessing (edge segmentation)	
$\delta_{\text{discont},L} = 10 \text{ cm}$	Distance threshold (LiDAR)
$\tau_{\text{sobel},S} = 128$	Sobel intensity threshold (stereo)
Plane segmentation	
$\delta_{\text{plane}} = 10 \text{ cm}$	Distance threshold
$\alpha_{\text{plane}} = 0.55 \text{ rad}$	Angular tolerance
$\delta_{\text{inliers}} = 10 \text{ cm}$	Distance threshold for outlier removal
Circle segmentation	
$\delta_{\text{circle},L} = 5 \text{ cm}$	Distance threshold (LiDAR)
$\delta_{\text{circle},S} = 1 \text{ cm}$	Distance threshold (stereo)
$\delta_{\text{radius}} = 1 \text{ cm}$	Radius tolerance (stereo)
$\delta_{\text{consistency}} = 6 \text{ cm}$	Geometry consistency tolerance
Clustering	
$N_{\text{cluster,min}} = \frac{1}{2}N'$	Minimum cluster size
$N_{\text{cluster,max}} = N'$	Maximum cluster size
$\delta_{\text{cluster}} = 5 \text{ cm}$	Cluster tolerance

A. Synthetic Test Environment

As stated before, the quantitative assessment of the set of extrinsic parameters relating two sensors in space is a nontrivial issue, as it is impossible, in practice, to obtain an exact ground-truth. Most works dealing with extrinsic calibration in the literature use manual annotations [15] or other approximations such as scene discontinuities [25].

In order to provide a comprehensive set of data describing the performance of the proposed method, we use the synthetic test suite proposed in [6], where the exact-ground truth of the relative transformation between sensors is available. The open-source Gazebo simulator [31] was used, and the operation modes of the three sensor modalities considered in this work (i.e., LiDAR, and stereo and monocular cameras) were faithfully replicated, taking into account the specifications of real devices in terms of field of view, resolution, and accuracy. In particular, Table II shows the set of devices used in the experiments.

TABLE II
SENSOR MODELS USED IN THE SYNTHETIC ENVIRONMENT

Device	Modality	Resolution ^a	HFOV
FLIR Bumblebee XB3	Stereo	1280 × 960	43°
Velodyne VLP-16	LiDAR	16 layers, 0.2°	360°
Velodyne HDL-32	LiDAR	32 layers, 0.2°	360°
Velodyne HDL-64	LiDAR	64 layers, 0.2°	360°
FLIR Blackfly S 31S4C-C	Monocular	2048 × 1536	85°

^a Image resolution, for cameras, and number of channels and horizontal (azimuth) angular resolution, for LiDAR scanners.

A model of the fiducial calibration target was also created by mimicking the appearance of the actual wooden embodiment shown in Fig. 2a. In the experiments, the target was placed with a wall behind so that LiDAR beams going through the circular holes reach a surface, generating the necessary gradient between foreground and background points.

Gaussian noise $\epsilon \sim \mathcal{N}(0, (K\sigma_0)^2)$ was applied to the sensors' captured data, with $\sigma_0 = 0.007 \text{ m}$ and $\sigma_0 = 0.008 \text{ m}$ for the pixel intensities and the LiDAR distances, respectively.

The noise factor K allows simulating ideal, noise-free environments ($K = 0$), realistic environments ($K = 1$), and noisy environments ($K = 2$). $K = 1$ is used by default.

Despite the eventual domain gap, experiments in this controlled setup enable systematic analysis and provide valuable insight into the method that will be otherwise unfeasible. Experimentation in the synthetic suite can be divided into three different focus points: reference point extraction, calibration with a single target position, and calibration with multiple target positions.

1) *Single-Sensor Experiments*: The first set of tests is aimed to analyze the accuracy in the extraction of the reference points from the four circular openings in the calibration target. Four different relative positions between sensor and calibration pattern, combining translations and rotations, were considered. Table III shows the position of the calibration pattern in sensor coordinates for each of these configurations, assuming that axes are defined as customary in LiDAR devices; i.e., x pointing forward, y to the left, and z upward. As in Sec. III, translation is denoted by (t_x, t_y, t_z) , whereas (r_x, r_y, r_z) represent roll, pitch, and yaw rotations (in radians).

TABLE III
RELATIVE SENSOR-TARGET POSES FOR REFERENCE POINT EXTRACTION ASSESSMENT

Cfg.	Translation (m)				Rotation (rad)		
	t_x	t_y	t_z	$ t $	r_x	r_y	r_z
P1	2.00	0.00	-0.50	2.06	0.0	0.0	0.0
P2	3.63	-0.50	-0.28	3.67	0.8	0.0	0.0
P3	5.38	-0.10	-0.50	5.41	0.0	-0.2	0.0
P4	6.50	-1.39	-1.43	6.80	0.0	0.0	-0.4

These setups were purposely chosen to investigate the limits of the reference point extraction branches. In fact, the method was unable to provide results in some extreme configurations; concretely, with the VLP-16 LiDAR in P3 and P4, the HDL-32 LiDAR in P4, and the stereo camera in P4 as well. In the case of the LiDAR scanners, their limited resolution made it impossible to find the circles at far distances, whereas the stereo was affected by the substantial degradation in depth estimation that this modality suffers as the distance increases. In typical use cases, it should be possible to avoid these situations by restricting the pattern locations to a reasonable range of distances with respect to the sensors.

The reference point localization performance was measured by determining the distance between the estimation provided by the approach and the ground-truth position of the center of the corresponding circle. The assignment was unambiguous in all cases and could be straightforwardly performed based on distance. Results were aggregated over three iterations for each pose and modality to account for the effect of the stochastic processes in the pipeline (e.g., RANSAC segmentations).

Firstly, Fig. 4 analyzes the effect of noise in the reference points location error. The results show that the procedure is highly robust to noise in all the modalities, given that the impact is limited to an increase in the standard deviation of the error in noisy situations ($K = 2$). In all cases, the error is well below 1 cm for the P1 and P2 configurations (upwards

and downwards triangle markers in the graph), whereas P3 (circle markers) and, especially, P4 (square markers) involve a significant increase across all the noise levels. This fact is particularly noticeable for the monocular modality (please note the different scale in the y-axis), where the accuracy in the detection of the ArUco markers proves to be much more sensitive to the size of their projections onto the image than to the pixel-wise noise.

Focusing on the realistic noise setup ($K = 1$), Fig. 5 shows the single-frame estimation error in each of the four configurations, further highlighting the relative position between sensor and calibration pattern as a significant factor. Apart from the most challenging configurations, the reference point localization proves accurate and precise across all the modalities, with LiDAR scanners exhibiting high robustness even in P3 and P4. As mentioned before, monocular struggles with these configurations but shows an excellent performance in P1 and P2.

The effect of the point aggregation and clustering strategy introduced in Sec. IV-E1 is investigated in Table IV, where the root-mean-square error (RMSE) of single-frame estimations and 30-iterations cluster centroids are compared under realistic noise conditions. The cluster centroid proves to be a consistently better representation of the reference points than the single-frame estimation in all cases, achieving a more remarkable improvement in situations with high dispersion; e.g., stereo in P3 (25.22% error reduction) or HDL-64 also in P3 (19.02% error reduction).

TABLE IV
RMSE (MM) IN REFERENCE POINT LOCATION USING A SINGLE-SHOT ESTIMATION (S) AND THE CLUSTER CENTROID AT $N = 30$ (C)

	P1		P2		P3		P4	
	S	C	S	C	S	C	S	C
Stereo	1.84	1.83	7.82	6.83	10.11	7.56	-	-
VLP-16	3.98	3.87	8.39	8.27	-	-	-	-
HDL-32	4.12	3.98	8.82	8.61	8.02	7.41	-	-
HDL-64	3.81	3.74	7.38	7.29	9.99	8.09	14.43	14.28
Mono	2.82	2.80	4.92	4.91	35.78	35.58	34.70	33.87

2) *Single-Pose Experiments*: Next, the full calibration pipeline will be evaluated considering only a single target position; that is, for $M = 1$. To that end, four combinations representative of real automotive sensor setups were analyzed:

- 1) HDL-32/HDL-64 (LiDAR/LiDAR)
- 2) Monocular/HDL-64 (camera/LiDAR)
- 3) Monocular/monocular (camera/camera)
- 4) Stereo/HDL-32 (camera/LiDAR)

Setups A and C embody situations where several devices of the same modality are included in the same sensor setup to enhance the field of view or the resolution of the captured data, whereas setups B and D exemplify setups aimed at camera/LiDAR sensor fusion. Both situations are frequently found in the onboard perception literature, even jointly on the same platform, e.g., [32].

For each setup, the three different relative positions between sensors reported in Table V were considered. They were

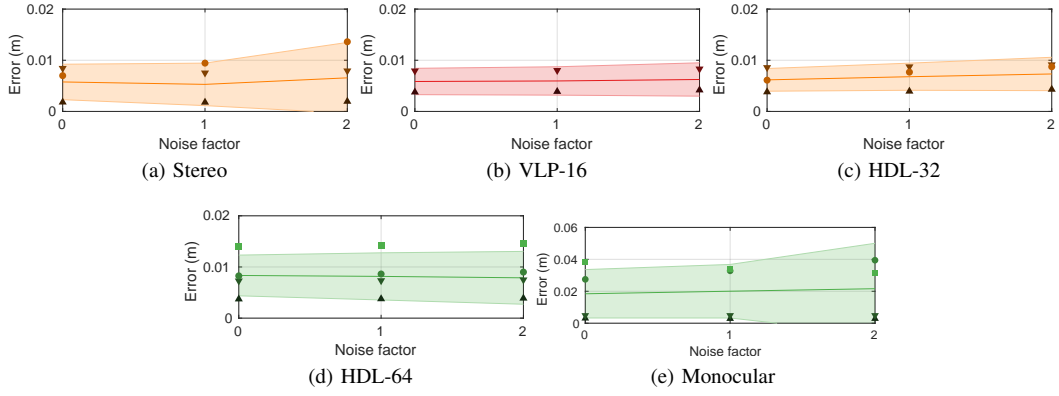


Fig. 4. Euclidean error in single-frame reference point localization vs. noise level (K), for each tested modality. The mean is depicted as a solid line, whereas the shaded area represents the standard deviation. Mean errors for each pose are depicted as individual markers (P1: upwards triangle, P2: downwards triangle, P3: circle, P4: square).

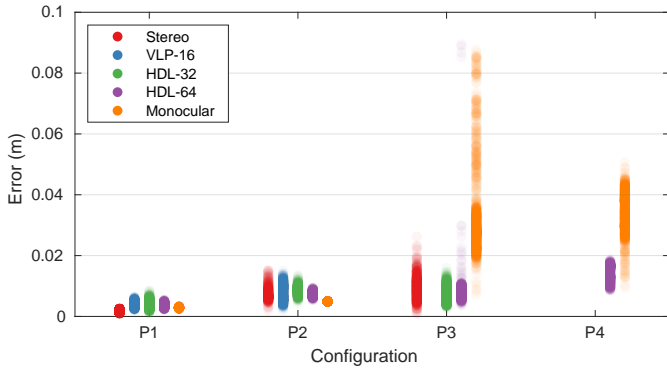


Fig. 5. Euclidean error in single-frame reference point localization, for each tested configuration and modality, with realistic noise.

picked from [6] as a representative set of configurations involving a wide range of translations and rotations. Representative pictures of these configurations in the synthetic test suite are depicted in Fig. 6. As in the previous case, three different iterations were considered in the results for each possibility. In all cases, the calibration pattern was placed arbitrarily in a location suitable for both sensors. Like in the per-sensor analysis, different distances to the target are used to further study its effect on final calibration.

TABLE V
TRANSFORMATION PARAMETERS OF THE DIFFERENT CALIBRATION SCENARIOS

Cfg.	t_x (m)	t_y (m)	t_z (m)	ψ (rad)	θ (rad)	ϕ (rad)
P1	-0.300	0.200	-0.200	0.300	-0.100	0.200
P2	-0.128	0.418	-0.314	-0.103	-0.299	0.110
P3	-0.433	0.845	1.108	-0.672	0.258	0.075

The analysis is now focused on the final calibration result. Therefore, following [15], results are given in terms of the linear (e_t) and angular (e_r) errors between the estimated rigid-body transformation and the ground-truth:

$$e_t = \|\hat{\mathbf{t}} - \mathbf{t}\| \quad (6)$$

$$e_r = \angle(\hat{\mathbf{R}}^{-1}\mathbf{R}) \quad (7)$$

Where \mathbf{t} is the translation vector, $\mathbf{t} = (t_x, t_y, t_z)$, and \mathbf{R} the 3×3 rotation matrix, representing the r_x , r_y , and r_z rotations; both elements compose the transformation matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad (8)$$

In the first place, the effect of the number of data frames used for reference point extraction, N , was studied. Fig. 7 aggregates the error for every setup and configuration when the calibration procedure is stopped at a point in the $N = [1, 40]$ interval. The results suggest that the method can provide a reliable estimation of the extrinsic parameters in a wide range of values of N , even with very few iterations. Nevertheless, $N = 30$ offers a fair accuracy-time tradeoff where outliers are extremely rare.

Table VI shows the linear (e_t) and angular (e_r) calibration errors sorted by sensor setup and configuration for $N = 30$. Monocular/monocular calibration (setup C) shows excellent accuracy and precision, in line with the reference point extraction results, featuring errors up to 100 times smaller than the rest of the setups. On the contrary, the stereo/HDL-32 (setup D) presents higher errors, likely due to the difficulties found by the stereo matching procedure to provide an accurate depth estimation at the distance where the pattern was placed in the experiments. Despite this, we observed that the implementation of the ArUco detector in use (OpenCV *aruco* module) was considerably more sensitive to light conditions than the stereo matching approach, so the method based on the stereo modality might still be useful in some instances. Overall, the results are reasonably accurate, even though the single-target situation poses a very challenging case for registration due to the coplanarity of the reference points, which can eventually become a source of ambiguity.

Table VII shows a comparison of the proposed approach with two single-pose LiDAR-camera calibration methods in the literature: the one by Geiger et al. [15], which estimates both the intrinsic and extrinsic parameters of the sensors with only one shot, and the one proposed by Velas et al. [16], which makes use of a calibration pattern very similar to ours. For a fair comparison, all the methods were fed with sensor data from the synthetic test suite, as reported in [6]. The sensor

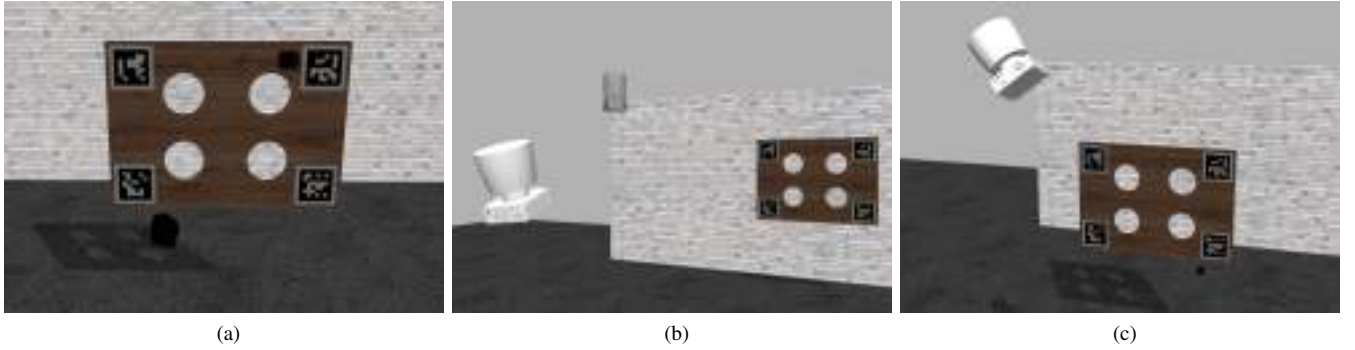


Fig. 6. Sensor setups for the single-pose experiments in the synthetic environment: P1 (a), P2 (b), and P3 (c)

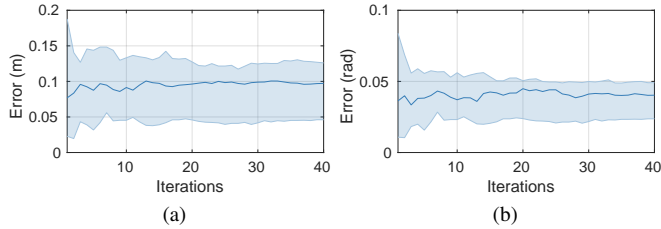


Fig. 7. Linear (a) and angular (b) calibration errors vs. number of iterations considered for clustering (N). The solid line represents the median and the shaded area, the interquartile range (IQR).

TABLE VI
MEAN (AND STANDARD DEVIATION) OF LINEAR (e_t) AND ANGULAR (e_r) CALIBRATION ERRORS FOR DIFFERENT SETUPS USING A SINGLE TARGET POSE ($M = 1$)

Set.	Error	P1	P2	P3
A	e_t (cm)	8.94 (1.49)	17.39 (2.13)	11.95 (1.56)
	e_r (10^2 rad)	4.36 (0.72)	3.91 (0.48)	5.80 (0.78)
B	e_t (cm)	10.34 (0.53)	4.31 (0.29)	9.68 (0.22)
	e_r (10^2 rad)	5.08 (0.26)	2.23 (0.13)	4.74 (0.12)
C	e_t (cm)	0.17 (0.01)	0.08 (0.00)	0.16 (0.00)
	e_r (10^2 rad)	0.03 (0.01)	0.04 (0.00)	0.04 (0.00)
D	e_t (cm)	9.62 (1.12)	47.02 (1.49)	31.60 (2.95)
	e_r (10^2 rad)	2.85 (0.34)	14.87 (0.47)	8.75 (0.84)

setup was composed of the stereo camera and the HDL-64 LiDAR introduced in Table II. We consider the two available options for reference point extraction in visual data: stereo and monocular, the latter employing the left image of the stereo rig as input. The errors were averaged over the same three poses used in the previous experiments.

TABLE VII
MEAN (AND STANDARD DEVIATION) OF LINEAR (e_t) AND ANGULAR (e_r) CALIBRATION ERRORS USING A SINGLE TARGET POSE ($M = 1$)

Method	e_t (m)	e_r (rad)
Geiger et al. [15]	0.93 (0.36)	1.30 (1.35)
Velas et al. [16]	0.99 (1.17)	0.35 (0.37)
Ours (Stereo-LiDAR)	0.12 (0.09)	0.04 (0.03)
Ours (Monocular-LiDAR)	0.12 (0.12)	0.04 (0.03)

According to these results, the stereo and mono alterna-

tives yield similar accuracy, significantly outperforming the other methods. Particularly noteworthy is the substantial improvement in angular error brought about by our approach, which stands out as the only one suitable for data fusion at far distances. These results prove that the baseline method, requiring a single pose of the calibration pattern ($M = 1$), works acceptably and provides a solid foundation for the full version with $M > 1$.

3) *Multi-Pose Experiments*: The last set of experiments focuses on the aggregation strategy presented in Sec. IV-E2, where the registration procedure is performed on $M \times 4$ points coming from M different calibration target positions. The sensor setups are identical to those used in the single-pose tests, but only the first configuration (P1) has been selected. For every sensor pair, the calibration pattern was moved along five different poses within a range of 5×5 m in front of the devices, up to 6 m in depth. To avoid the eventual bias introduced by the order in which these positions are used, results are obtained through three different iterations in which the sorting is changed.

The evolution of the linear and angular calibration errors with M follows an almost-exponential decay for all the tested setups, as shown in Fig. 8 (please note the logarithmic scale). Only by introducing an additional target pose, an average reduction of 61.2% (linear) / 68.15% (angular) can be achieved. When five poses are employed, the errors drop by 85.42% (linear) / 87.01% (angular). The largest decreases correspond to the HDL-32/HDL-64 setup, where the reduction is around 97% for both kinds of errors, yielding a final calibration with a deviation of 6.5 mm and 0.002 rad from the ground-truth.

The proposed approach has been compared with the state-of-the-art method recently introduced by Zhou et al. [17], aimed at LiDAR-camera calibration using one or several views of a checkerboard. To that end, we used the implementation included in the MATLAB Lidar Toolbox [33]. Tests were performed with the monocular/HDL-64 sensor setup, using $M = 2$ and $M = 3$ poses of the respective calibration patterns. Mean calibration errors by both methods are shown in Table VIII.

As apparent from the results, the performance of both approaches is comparable, although our method achieves consistent improvements that even exceed 50% for the angular error when $M = 3$. These results confirm the effectiveness

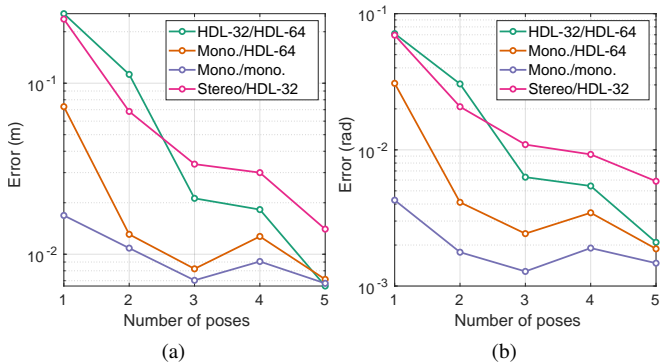


Fig. 8. RMSE of the linear (a) and angular (b) calibration errors (m and rad) vs. number of calibration poses (M) for four sensor setups.

TABLE VIII
MEAN OF LINEAR (e_t) AND ANGULAR (e_r) CALIBRATION ERRORS USING SEVERAL TARGET POSES ($M > 1$)

Method	$M = 2$		$M = 3$	
	e_t (cm)	e_r (10^2 rad)	e_t (cm)	e_r (10^2 rad)
Zhou et al. [17]	1.51	0.63	1.08	0.50
Ours	1.15	0.39	0.82	0.24

of the aggregation of reference points across different target locations, providing a calibration solution that features sub-centimeter accuracy.

B. Real Test Environment

The set of experiments presented in the previous section offers a systematic and exact analysis of the performance of the proposed calibration method. Nevertheless, experiments in a real use case were also carried out to validate the applicability of the approach, assessing its adequacy to meet the requirements of the intended application.

Two VLP-16 LiDARs and a Bumblebee XB3 camera were mounted in a rig, with different rotations emulating the ones that can be found in vehicle setups. The suite was then calibrated using the proposed method and placed on the roof of an autonomous vehicle [34], thus allowing the qualitative assessment of the results in real traffic scenarios.

As our method is intended for extrinsic calibration of pairs of sensors, the procedure involved two independent calibrations: camera/LiDAR and LiDAR/LiDAR. The monocular alternative of the reference point extraction process was employed, together with the one introduced for the LiDAR modality. Note that the low number of layers of the LiDAR scanners pose an additional challenge, as the set of locations where the four circles of the calibration pattern are fully visible is restricted. As with the synthetic experiments, points were extracted from the accumulation of $N = 30$ frames, and $M = 5$ target poses were used. The rest of the parameters remained unchanged from Table I.

Ground-truth of the relative position between sensors was not available, but some illustrative statistics about the performance of the calibration procedure with real sensors are presented in Fig. 9. On the one hand, Fig. 9a shows the dispersion

of the estimated reference points across different poses of the calibration pattern, each represented by a point. Data from the two separate calibration procedures are included. The black line represents the mean, the dark shadow spans the standard deviation, and the light shadow covers 1.96 times the standard error of the mean.

The results confirm that the dispersion in the LiDAR modality is significantly higher than the one exhibited by its monocular counterpart, as suggested by the tests in the synthetic environment. However, the deviation is still small enough to enable higher accuracy in registration.

On the other hand, Fig. 9b shows the difference, measured in linear and angular errors, of the calibrations performed with $M \in [1, 4]$ versus the final result with $M = 5$. The results validate the conclusion drawn in the previous section: using several pattern poses ($M > 1$) causes significant changes in the calibration result up to 3 poses, where it plateaus.

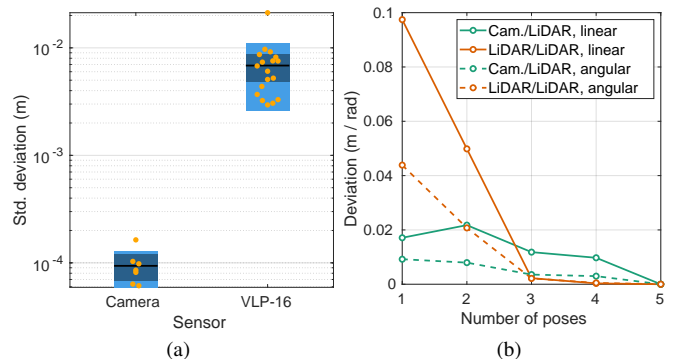


Fig. 9. Statistics from real sensor data: Dispersion in the localization of the reference points (a) and calibration deviation from the final result (linear, in m, and angular, in rad) (b).

Finally, Fig. 10 depicts four examples of traffic scenarios captured by the calibrated sensor setup, with specific regions enlarged so that the details can be well perceived. As shown, the use of the extrinsic parameters extracted by the proposed approach enables a perfect alignment between both data modalities, even at a considerable distance from the car, being especially noticeable when representing thin objects (e.g., lamp poles or trees).

VII. CONCLUSION

We have presented an approach to obtain the extrinsic parameters representing the relative pose of any pair of sensors involving LiDARs, monocular or stereo cameras, of the same or different modalities. Unlike the existing works, the simplicity of the calibration scenarios and the characteristics provided by the proposed target allow obtaining accurate results for most sensing setups featured by autonomous vehicles. Moreover, no user intervention is required during the calibration process.

Additionally, we have introduced an advanced simulation suite that copes with the traditional imprecision at calibration performance assessment and provides an exact ground truth that enables a reliable evaluation of extrinsic calibration methods.

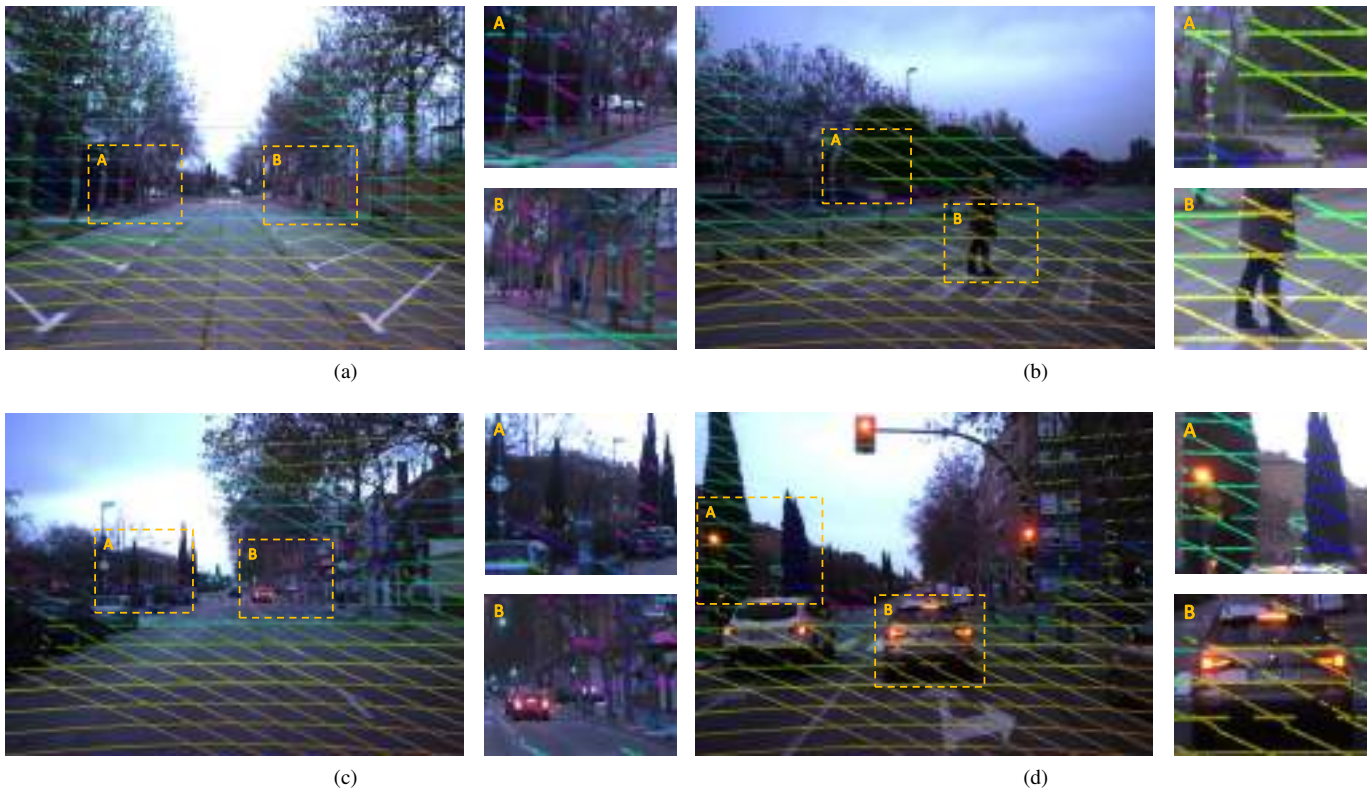


Fig. 10. Samples (main view and two close-up views) of different traffic scenarios where LiDAR points have been projected onto the image using the set of extrinsic parameters extracted with the proposed approach.

Results obtained from the conducted experiments demonstrate that the algorithm presented in this work notably outperforms existing approaches. Tests performed over real data confirm the accuracy obtained in the simulation environment.

Different lines of work remain open for the future. An outlier rejection scheme might be useful to discard spurious samples obtained in the reference point extraction procedure. At this point, accurate modeling of the sensor noise could be convenient, which will also enable adapting the parameter settings to each particular device. On the other hand, the proposed method has been designed to determine a fixed set of extrinsic parameters before the perception system is deployed; however, sensor setups mounted in movable platforms, such as autonomous vehicles, can suffer miscalibrations during regular operation. The use of the proposed method would require the ability to detect these situations early, prompting the user to perform a recalibration when necessary.

The proposed approach provides a practical and effective solution to a classic problem in autonomous driving and robotics in general, such as the calibration of different sensor units, which had not been solved yet. Although there is still a road ahead, this proposal solves a common problem for the scientific community working in this field, bringing autonomous driving and robotics solutions closer to their final deployment.

The implementation of the calibration method described in this paper has been made publicly available to promote reproducibility and provide researchers and practitioners in

the field with a tool to face the usual problem of extrinsic calibration in an easy and effective way. The software makes use of open source libraries (e.g., OpenCV and PCL) and is available as a package in the popular ROS framework¹. The synthetic test suite used for the experimentation has been also released².

REFERENCES

- [1] U. Franke, D. Pfeiffer, C. Rabe, C. Knoepfel, M. Enzweiler, F. Stein, and R. G. Hertwich, "Making Bertha see," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, dec 2013, pp. 214–221.
- [2] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, F. Duffhauß, C. Gläser, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [3] J. Beltrán, C. Guindel, I. Cortés, A. Barrera, A. Astudillo, J. Urdiales, M. Álvarez, F. Bekka, V. Milanés, F. García, and M. Alvarez, "Towards autonomous driving: a multi-modal 360° perception proposal," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020.
- [4] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 918–927.
- [5] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5750–5757.
- [6] C. Guindel, J. Beltrán, D. Martín, and F. García, "Automatic extrinsic calibration for lidar-stereo vehicle sensor setups," in *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 674–679.

¹http://wiki.ros.org/velo2cam_calibration

²https://github.com/beltransen/velo2cam_gazebo

- [7] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 4164–4169.
- [8] S. Debattisti, L. Mazzei, and M. Panciroli, "Automated extrinsic laser and camera inter-calibration using triangular targets," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 696–701.
- [9] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3D LIDAR instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.
- [10] M. Pereira, D. Silva, V. Santos, and P. Dias, "Self calibration of multiple LIDARs and cameras on autonomous vehicles," *Robotics and Autonomous Systems*, vol. 83, pp. 326–337, 2016.
- [11] Z. Pusztai and L. Hajder, "Accurate calibration of LiDAR-camera systems using ordinary boxes," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 394–402.
- [12] Y. Li, Y. Ruichek, and C. Cappelletti, "3D triangulation based extrinsic calibration between a stereo vision system and a LIDAR," in *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 797–802.
- [13] S. Verma, J. S. Berrio, S. Worrall, and E. Nebot, "Automatic extrinsic calibration between a camera and a 3D lidar using 3D point and plane correspondences," in *Proc. IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3906–3912.
- [14] A. Dhall, K. Chelani, V. Radhakrishnan, and K. Krishna, "LiDAR-camera calibration using 3D-3D point correspondences," *arXiv:1705.09785 [cs.RO]*, May 2017.
- [15] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3936–3943.
- [16] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of RGB camera with Velodyne LiDAR," in *Comm. Papers Proc. International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2014, pp. 135–144.
- [17] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3D LiDAR using line and plane correspondences," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5562–5569.
- [18] J. Jiao, Q. Liao, Y. Zhu, T. Liu, Y. Yu, R. Fan, L. Wang, and M. Liu, "A novel dual-lidar calibration algorithm using planar surfaces," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1499–1504.
- [19] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3685–3691.
- [20] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-lidar calibration: A targetless and structureless approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1556–1563, 2020.
- [21] C. H. Rodríguez Garavito, A. Ponz, F. García, D. Martín, A. de la Escalera, and J. M. Armingol, "Automatic laser and camera extrinsic calibration for data fusion using road plane," in *IEEE International Conference on Information Fusion (FUSION)*, 2014.
- [22] L. Tamas, R. Frohlich, and Z. Kato, "Relative pose estimation and fusion of omnidirectional and lidar cameras," in *Computer Vision - ECCV 2014 Workshops*, 2014, pp. 640–651.
- [23] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1803–1810.
- [24] G. Iyer, K. R. R., J. K. Murthy, and K. M. Krishna, "CalibNet: Self-supervised extrinsic calibration using 3D spatial transformer networks," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1110–1117.
- [25] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," in *Robotics: Science and Systems*, 2013.
- [26] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.
- [27] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [28] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.
- [29] R. Szeliski, *Computer vision: algorithms and applications*, 1st ed. New York, New York, USA: Springer, 2010.
- [30] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.
- [31] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 2149–2154.
- [32] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 621–11 631.
- [33] The Mathworks, Inc. MATLAB Documentation: Lidar-camera calibration. [Online]. Available: <https://www.mathworks.com/help/lidar/lidarcameracalibration.html>
- [34] M. Á. de Miguel, F. M. Moreno, P. Marín-Plaza, A. Al-Kaff, M. Palos, D. Martín, R. Encinar-Martín, and F. García, "A research platform for autonomous vehicles technologies research in the insurance sector," *Applied Sciences*, vol. 10, no. 16, p. 5655, 2020.



Jorge Beltrán received his his M.Sc degree in Robotics and Automation from Universidad Carlos III de Madrid, Spain, in 2016. He is currently pursuing his Ph.D. in Electrical, Electronic and Automatic Engineering at the same University. Since 2014, he is part of the Intelligent Systems Laboratory, where he worked as a Research Assistant. His research interest includes sensor calibration, deep learning, and sensor fusion for object detection in the field of automated vehicles.



Carlos Guindel received his Ph.D. degree in Electrical, Electronic and Automatic Engineering from Universidad Carlos III de Madrid (Spain) in 2019. He is with the Intelligent Systems Laboratory (LSI) research group since 2011. His research interests include computer vision and deep learning, with a focus on their application to environment perception in automated vehicles.



Fernando García received his Ph.D. degree in Electrical, Electronic and Automatic Engineering from Universidad Carlos III de Madrid in 2012 where he works as Associate Professor. His research interests are perception and data fusion, mainly applied to vehicles and robotics. He is member of the Board of governors of the IEEE-ITS Society since 2017 and chair of the Spanish chapter for the period 2019-2020.