Spatiotemporal Calibration of Camera and 3D Laser Scanner

Michał R. Nowicki[1] iD

Abstract— The multi-sensory setups consisting of the laser scanners and cameras are popular as the measurements complement each other and provide necessary robustness for applications. Under dynamic conditions or when in motion, a direct transformation (spatial calibration) and time offset between sensors (temporal calibration) is needed to determine the correspondence between measurements. We propose an open-source spatiotemporal calibration framework for a camera and a 3D laser scanner. Our solution is based on commonly available chessboard markers requiring one-minute calibration before the operation that offers accurate and repeatable results. The framework is based on batch optimization of point-to-plane constraints with a time offset calibration possible by a novel minimal, continuous representation of the plane equations in the Lie algebra and the use of B-splines. The framework's properties are evaluated in simulation while performance is verified with two distinct sensory setups with Velodyne VLP-16 and SICK MRS6124 3D laser scanners.

I. INTRODUCTION

Nowadays, the sensory setups of mobile robots consist of an increasing number of cameras, LiDARs, radars, and AHRS units to provide complementary information used by autonomous robots. To use different sensors for the state estimation it is necessary to represent the measurement from one sensor in the coordinate system of the other sensor. Under static conditions, only the spatial transformation is needed. But when sensors are in motion (or when the environment is changing), the knowledge on the temporal relationship between sensors is necessary.

The spatial calibration can be estimated with already existing frameworks or assumed based on CAD drawings. The temporal calibration is usually performed with the additional hardware that triggers cameras and timestamps acquired 3D laser scans. This additional hardware is often not available out-of-the-box and some cheaper versions of sensors do not offer such possibility at all.

In our work, we target the problem of joint spatiotemporal calibration of a sensory system consisting of a camera and a 3D laser scanner (LiDAR), as presented in Fig. 1A. The spatial calibration provides a possibility to determine the correspondence between measurements from sensors under static conditions but fails in a dynamic environment (Fig. 1B) while spatiotemporal calibration provides expected results (Fig. 1C). Spatiotemporal calibration is a necessity for the
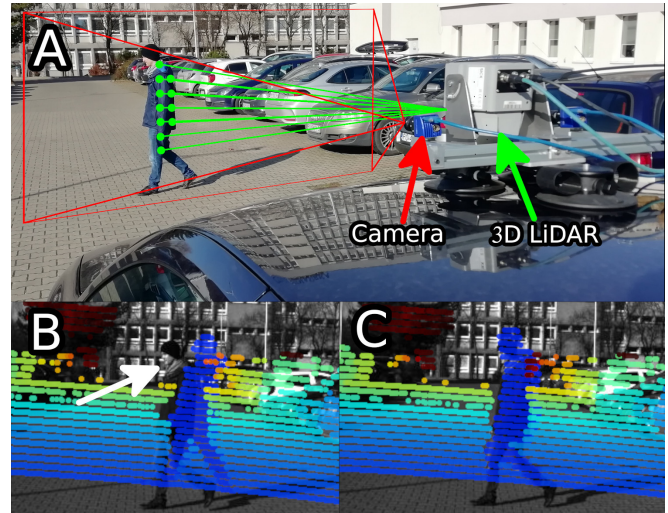
Fig. 1: Common observation from a camera-3D LiDAR sensory setup mounted on a car (A) and LiDAR points projected onto the image in a dynamic scenario when spatial (B) or spatiotemporal (C) calibration was performed with the proposed framework. Notice how lack of temporal calibration leads to incorrect sensor correspondence (white arrow)

currently popular tightly integrated Visual Odometry (VO) or Simultaneous Localization and Mapping (SLAM) solutions that jointly process data from a camera and 3D LiDAR [1].

The proposed calibration is based on the motion of a chessboard pattern and a sequence of successive observations from the camera and 3D LiDAR. The system performs batch optimization of constraints to estimate 6 DoF (degree-of-freedom) transformation between sensors and the corresponding time offset. The optimization is based on independent LiDAR's point to the camera's plane constraints that are extended by the continuous-time plane representation to allow time offset optimization.

The contribution of our work can be summarized as:

- spatiotemporal camera - 3D laser scanner calibration that makes it possible to estimate 6 DoF transformation and the time offset between sensors using common chessboard marker,
- novel continuous-time minimal plane representation using the Lie algebra with B-splines,
- calibration solution that converges to the real spatial and temporal relationship even with poor initial guesses,
- publicly available, ROS-compatible calibration frame-

work[1].

II. RELATED WORK

The need for multisensor calibration can be dated back to camera-2D laser sensory rigs that were used to design multisensor SLAM systems [2]. The first popular and publicly available framework for spatial calibration between a camera and a 2D laser scanner was presented by Zhang and Pless in [3]. The calibration was a result of an optimization performed over several observations from both sensors detecting a camera calibration pattern in discrete calibration poses. The 2D laser points lying on the camera calibration plane were used to form and solve the non-linear optimization problem with point-to-plane constraints.

The first spatial calibration of a camera and 3D LiDAR was presented by Scaramuzza in [4]. This solution was based on manual markings between corresponding points between the camera's image and LiDAR's scan converted to range images and was suited to rotating 2D laser scanners under static conditions that could generate high-resolution range images.

The spatial calibration of a camera and 3D LiDAR was later tackled with the use of specially modified calibration patterns with four circular holes [5], a box with 3 perpendicular sides (trihedron) [6], or just regular boxes [7]. Despite different approaches to detection, constraint extraction, and optimization formulation, none of these approaches provides temporal calibration.

More recently, another branch of solutions emerged with an idea to perform calibration based on a comparison between the independent trajectories from both (or multiple) analyzed sensors. These solutions can also be based on tracking of some object (marker), like a ball of a certain color in the calibration of multiple RGB-D Kinect v2 sensors [8]. The comparison can also be made marker-less, like proposed by [9] that focuses on working with short sequences or [10] that considers sensor's accuracy to provide the uncertainty of the obtained calibration. In [11], the authors focus on the trajectory representation proposing Gaussian processes as a better alternative for trajectory alignment. The markerless solutions do not rely on the placement of an artificial marker in the environment but the experimental environment is expected to be well-suited for VO/SLAM systems.

The spatiotemporal calibration with those approaches is usually performed by decoupling the problem of spatial and temporal calibration. The temporal calibration is often found with some form of cross-correlation on the time-evolution of a signal that is independent of the spatial calibration (like linear or rotational speeds) [10], [11]. Some researchers believe that these estimations should not be decoupled [12] and the problem should be solved simultaneously. More recently, Park et al. [13] proposed a solution that utilizes linear interpolation on the $SE(3)$ Lie group to jointly solve the spatiotemporal problem for marker-less calibration.

We propose a spatiotemporal calibration that is marker-based and relies on continuous relative motion between the sensory setup and the marker. The continuous motion provides us with a significantly bigger number of measurements than the static calibration while a novel continuous-time representation of plane equations in the Lie algebra with B-splines makes it possible to estimate time offsets between sensors. Compared to marker-less solutions, our marker-based solution is independent of the accuracy of the VO/SLAM or the type of the environment, while also providing both spatial and temporal calibration. Due to the continuous nature of data acquisition, the proposed calibration is similar to the *kalibr* software [14], that offers spatiotemporal calibration for multi-camera and multi-AHRS sensory setups, where time-varying states are represented by B-spline functions, as introduced in [15]. Unfortunately, *kalibr* only provides an experimental camera-LRF (2D LiDAR) calibration that is no longer maintained in the public repository. Therefore, there is a need for our contribution that can be combined with *kalibr* under ROS to calibrate more complex sensory setups.

III. PROPOSED SOLUTION

The proposed framework is dedicated to sensory setups with global-shutter cameras and 3D LiDARs that are rigidly attached. We assume that the camera is internally calibrated and these parameters are fixed during the calibration. Besides, we assume that the time offset between sensors is unknown, small, and is approximately constant. This assumption is true in most cases, especially when the sensors are connected to the same computer.

The calibration procedure consists of motion of a calibration pattern that should be jointly and continuously observed by the camera and by the 3D LiDAR. Depending on the preference of the user, either the sensory setup or the calibration pattern can be moved. After recording, we process the data as presented in Fig. 2. The calibration pattern is detected on the camera images using typical chessboard detection algorithm from OpenCV. The LiDAR points belonging to the plane of the calibration pattern are detected using range images either by manual marking of the corners or with a semi-automatic tracking procedure from the previous detection. Without temporal calibration, the spatial calibration can be determined by the point-to-plane optimization based on a discrete number of calibration poses:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \sum_i \sum_j \boldsymbol{\pi}(t_i)^\top \mathbf{T} \mathbf{p}_i, \qquad (1)$$

where $\mathbf{T}^*$ is the sought transformation from the coordinate system of the LiDAR to the coordinate system of the camera, $\mathbf{p}_i$ denotes the homogeneous coordinates of the i-th 3D laser point on the calibration pattern, and $\boldsymbol{\pi}(t_i)$ is the plane equation of the chessboard pattern estimated by the camera at timestamp $(t_i + \Delta t)$. The plane $\boldsymbol{\pi}(t_i)$ can be determined by finding the plane equation for the detection on the camera image with the closest timestamp. The frequency of the
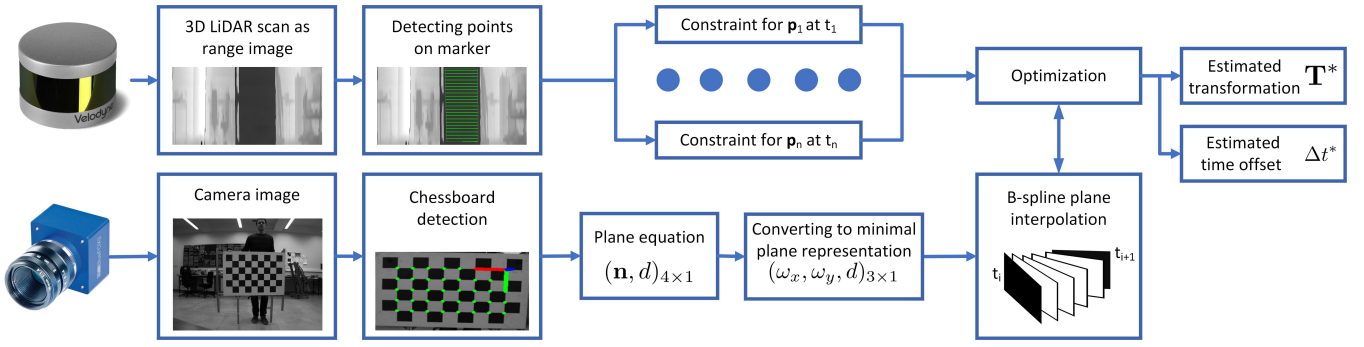
Fig. 2: The overall processing steps of the proposed spatiotemporal camera-3D LiDAR calibration

captured data from both systems can be different in the presented formulation.

In the proposed framework, we are also concerned with temporal calibration, which raises several issues as data from both sensors are not recorded in the same moments (timestamps) and 3D LiDARs are constantly rotating, which results in motion distortions similar to those in rolling-shutter cameras. These distortions can be compensated to create a point cloud at selected timestamp within SLAM solutions, like for 3D LiDARs in [16] or cameras in [17], but such solutions inevitably introduce additional errors. We opt to treat each point with the corresponding timestamp independently, similarly to the first odometry steps in [16] or in [18]. In this more complicated case, the spatiotemporal calibration is determined by the following optimization:

$$\mathbf{T}^*, \Delta t^* = \arg\min_{\mathbf{T}, \Delta t} \sum_i \boldsymbol{\pi}(t_i + \Delta t)^\top \mathbf{T} \mathbf{p}_i, \qquad (2)$$

where Δt^* is the estimated time difference between timestamps of the camera and the 3D LiDAR, $\boldsymbol{\pi}(t_i + \Delta t)$ is the plane equation of the chessboard pattern estimated by the camera at timestamp $(t_i + \Delta t)$, and t_i is the acquisition timestamp of the i-th laser point $\mathbf{p}_i$ lying on the plane of the calibration pattern. Formulation of the problem as in (2) requires individual timestamps for each 3D laser point and the knowledge of the plane equation at any arbitrary timestamp. The optimization is performed using the Levenberg-Marquardt algorithm with the Huber robust cost available in the *g2o* library [19].

A. Time offsets for points from 3D laser scanner

Most of the available 3D laser scanners operate as continuously rotating devices that measure the distance based on the time-of-flight principle. As each device is constructed differently, the direct timestamps for each laser point have to be determined. The best calibration results should be expected if sensor drivers provide real timestamps for all considered points. However, in some case it is not possible and the real laser point timestamps have to be estimated using the available information about the method of acquiring individual measurements in the particular LiDAR type. We present laser point timestamps estimation for Velodyne VLP-16 and SICK MRS6124 that have very different patterns of laser points acquisition.
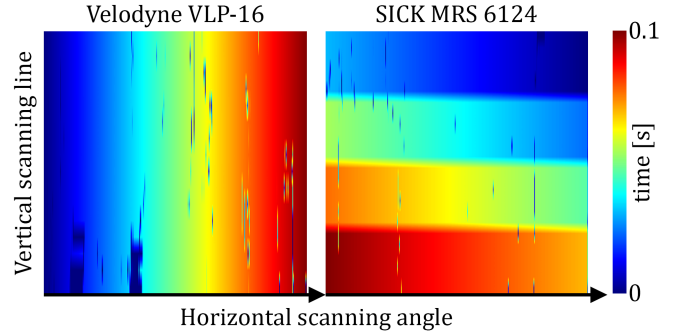

Fig. 3: Time offsets for laser points belonging to Velodyne VLP-16 and SICK MRS6124 depending on the time of acquisition due to beam rotation. In real scans, some distance measurements are invalid and thus some missing time offsets can be observed, visible as artifacts on the presented time offset images

The Velodyne VLP-16 has 16 scanning layers, 360° field of view, and in the typical configuration rotates at $f = 10$ Hz in the clockwise direction. For each retrieved point cloud, we determine the starting (lowest) angle ϕ_s and the ending (largest) angle ϕ_e as the angles on the horizontal scanning plane. The time offset t_i for the point $\mathbf{p}_i$ is computed as:

$$t_i = t_{\text{cloud}} + \frac{\phi_i - \phi_s}{f(\phi_e - \phi_s)}, \qquad (3)$$

where t_{cloud} stands for the timestamp of the pointcloud and ϕ_i is the horizontal scanning angle for the analyzed point.

The SICK MRS6124 has 24 scanning layers but these layers are formed from 4 groups with 6 independent scanning lines. The sensor has a 120° field of view and rotates in the counter-clockwise direction. This scanner also provides data with $f = 10$ Hz, but rotates 4 times faster as the 6 independent lines perform full scanning for each group one after the other. In this case, the time offset t_i for point p_i is computed as:

$$t_i = t_{\text{cloud}} + \frac{1}{f}\left(\frac{g}{4} + \frac{\phi_i - \phi_s}{2\pi}\right), \qquad (4)$$

where g is the number of the scanning group (0, 1, 2 or 3) and $\phi_s = \frac{\pi}{3}$.

The time offsets for points in the same scan for both sensors are visualized in Fig. 3.

B. Continuous plane representation

The chessboard pattern is detected in each incoming camera image. With the knowledge of the camera parameters and real chessboard size, the transformation between the camera coordinate system and coordinate system of the chessboard pattern is determined. Then, the chessboard calibration plane equation is computed and represented by the 4-dimensional vector $(\mathbf{n}, d)$, where $\mathbf{n}$ is the normalized three-dimensional plane normal, and d is the distance to the origin of the coordinate system. From all images, we get a set of plane equations observed at discrete timestamps of images.

To determine the plane equations at the timestamps between timestamps of images, we need to interpolate these plane equations. The 4-dimensional representation is not minimal and thus the interpolation might have resulted in plane equations that require further re-normalization. To avoid that issue, we propose a minimal 3-dimensional representation utilizing properties of the $SO(3)$ group (following the ideas in [20]) and its corresponding Lie algebra elements. The exponential map (exp) transforms elements from Lie algebra tangential space ($\mathfrak{so}(3)$) to $SO(3)$ and its inverse logarithmic map $\log$ in order to perform inverse transformation (from $SO(3)$ to $\mathfrak{so}(3)$). The idea is that only two parameters are needed to represent a normalized vector (or a point on unit hemisphere). In the given context of plane equation interpolation between camera frames, the proposed method provides a robust and accurate approximation of the mathematically rigorous iterative interpolation on the sphere manifold S^2 [21].

The overparametrization of a default plane representation stems from the fact that the plane normal $\mathbf{n} = (n_x, n_y, n_z)$ is normalized, and thus can be represented with two components of the Lie algebra of the $SO(3)$ group $(\omega_x, \omega_y, 0)$. Following equations of the logarithmic map $\log$ for the special case with $\omega_z = 0$, we derived equations:

$$\theta = \mathrm{acos}(n_z), \tag{5}$$

$$\omega_x = -n_y \frac{\theta}{\sin(\theta)}, \quad \omega_y = n_x \frac{\theta}{\sin(\theta)}, \tag{6}$$

that are used in conversion to minimal representation $((\mathbf{n}, d)_{4\times1} \rightarrow (\omega_x, \omega_y, d)_{3\times1})$. Special care has to be taken when $\theta \rightarrow 0$ (we perform series expansion of $\frac{\theta}{\sin(\theta)}$) and when $\theta \rightarrow \pi$ (we avoid the issue by assuming equivalent plane representation $\mathbf{n}' = -\mathbf{n}$ and $d' = -d$). If necessary, the original, 4-dimensional representation can be retrieved with $((\omega_x, \omega_y, d)_{3\times1} \rightarrow (\mathbf{n}, d)_{4\times1})$:

$$\mathbf{n} = \exp([\omega_x, \omega_y, 0]) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{7}$$

where $\exp(\cdot)$ computes the Lie group representation based on the Lie algebra element. More information about the usage of Lie algebra in robotics can be found in [22].

C. Plane equation interpolation

The minimal plane representation makes it possible to determine plane equations at required timestamps by converting to the minimal representation, performing interpolation, and then returning to the usual 4-dimensional plane equation. For this task, we tried linear interpolation of parameters, but the lack of continuous derivatives is an important issue that results in optimization getting stuck in local minima. Therefore, we propose to perform interpolation with cubic B-splines for continuous plane representations that guarantee continuous first and second derivatives of the minimal plane representation.

In our formulation of the B-spline we use the cumulative form as presented in [23]:

$$\mathbf{s}(t) = \mathbf{s}_0 B_0(t) + \sum_{i=1}^{n}(\mathbf{s}_i - \mathbf{s}_{i-1})B_i(t), \tag{8}$$

where $\mathbf{s}(t)$ is the interpolated value at time t, $\mathbf{s}_i$ is the value of the i-th control point that is a known value from a measurement, $B_i(t)$ is the i-th component of the cumulative basis function, and n is the order of the B-spline. The same formulation for the Lie algebra elements can be written as [24]:

$$\mathbf{r}(t) = \log\{\exp(\mathbf{r}_0 B_0(t)) \prod_{i=1}^{n} \exp(\mathbf{\Omega}_i B_i(t))\}, \tag{9}$$

where $\mathbf{r}(t)$ is the interpolated value of the Lie algebra element at time t, $\mathbf{r}_i$ is the value of the i-th control point in the Lie algebra (known value for time t_i), $\mathbf{\Omega}_i = \log\left(\exp(\mathbf{r}_{i-1})^T \exp(\mathbf{r}_i)\right)$ is the equivalent of a difference for Lie algebra elements. We use cubic B-spline ($n = 4$) and thus the cumulative basis function is equal to:

$$\mathbf{B}(u) = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}, \tag{10}$$

where u is the normalized time (from 0 to 1) for the considered interval between t_2 and t_3. In practice, we also check that the timestamps of the detected chessboards (t_1, t_2, t_3, t_4) are uniformly distributed in time and do not perform interpolation if this condition is not satisfied.

In the proposed solution, with the minimal plane representation (ω_x, ω_y, d), we follow (8) for the interpolation of element d and (9) for Lie algebra components (ω_x, ω_y). Importantly, the analytical Jacobians of the B-spline can be derived for the presented interpolation using the well-known equations from [24]. However, our system uses the more efficient analytical Jacobians presented in [25].

IV. ANALYSIS OF THE CALIBRATION SYSTEM'S PROPERTIES

A. Simulation

We evaluated our system in a series of Monte-Carlo type simulation experiments. Each experiment starts with the creation of a motion that resembles a real movement of the

calibration marker in front of a sensory setup and lasts for 50 seconds. We start by randomly sampling 11 poses of the calibration marker in the typical working area of a calibration (cuboid of size $8 \times 2 \times 4$ meters) with an angle between plane normal and camera optical axis not exceeding $90°$. These 11 poses define control points for the simulated trajectory at timestamps 0, 5, 10, ..., 50 s. To achieve continuous motion of the calibration marker in the environment between these poses, we interpolate positions of the four corners of the calibration marker using B-splines on the $SE(3)$ group. Based on the motion of the chessboard marker, we simulate camera chessboard detections with the desired framerate (10 Hz) and compute LiDAR points that would lie on that calibration pattern assuming random camera-LiDAR transformation with $t_x \in (-1, 1), t_y \in (-0.5, 0.5), t_z \in (-0.25, 0.25)$ meters and angular rotation up to $45°$ by random axis. In the end, we achieve a set of chessboard detections at selected timestamps and a set of LiDAR points with corresponding timestamps. The initial guess of transformation is computed as a modification to the real transformation with $t_x \in (-0.1, 0.1), t_y \in (-0.1, 0.1), t_z \in (-0.1, 0.1)$ meters and angular rotation up to $22.5°$ by random axis. The time offset initial guess is set to 0.

B. Influence of LiDAR's accuracy

In the simulation, we assumed that the 3D LiDAR is internally calibrated with a range measurement that has a Gaussian noise [26]. We performed 1900 random experiments (100 different trajectories for the motion of the calibration marker for 19 different time offsets) for each assumed standard deviation of the Gaussian noise of the LiDAR range measurements (σ_{LiDAR}). In each case we measured the metric, rotation, and time offset estimation errors between the real and the estimated values (Fig. 4).
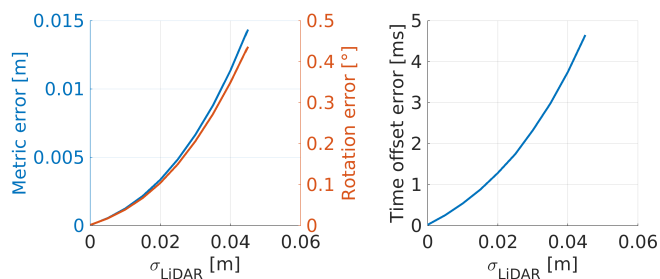

Fig. 4: Metric, rotational and time offset errors of the calibration depending on the standard deviation of the LiDAR range measurement (σ_{LiDAR})

The errors grow significantly when σ_{LiDAR} increases with a measured average error of 1.13 cm, $0.35°$ and 3.75 ms when σ_{LiDAR} is equal to 0.04 m. Fortunately, the Velodyne VLP-16 has an accuracy of ± 2.5 cm, which corresponds to the standard deviation of about 0.01 m. In such a case, we obtained average calibration errors of 0.12 cm, $0.04°$ and 0.54 ms.

C. Influence of the initial time offset guess

A good initial guess is crucial for gradient-based optimization to converge to the global minimum. In real-world scenarios, the sensible initial guess for spatial transformation can usually be provided (i.e. by measuring with a tape), which might not be true for temporal calibration (although cross-correlation can be used in many scenarios). The results for experiments with initial time offsets between sensors that differed from -90 ms to 90 ms from ground truth time offsets are presented in Fig. 5.
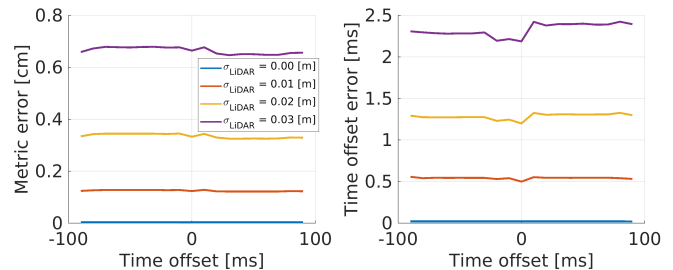

Fig. 5: Metric error and time offset error depending on the initial time offset guess and chosen standard deviation of the LiDAR range measurement (σ_{LiDAR})

In each considered case, our system converged to similar metric and time offsets regardless of the initial time offset guess and initial guess of the transformation proving empirically that the optimized function is convex. The metric and time offset errors increase with standard deviation values σ_{LiDAR}.

D. Spatial vs. spatiotemporal calibration

We performed experiments when only the spatial transformation between sensors was optimized to check what errors are to be expected if no temporal calibration would be performed (Fig. 6).
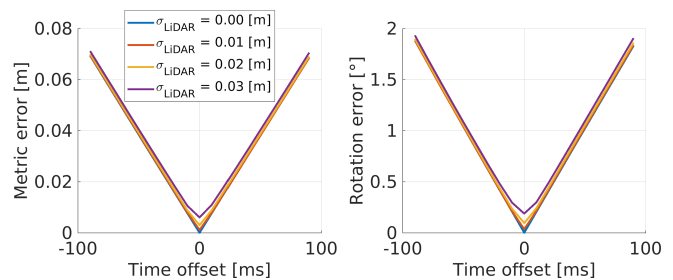

Fig. 6: Metric, rotational and time offset errors of the calibration depending on the chosen time offset when only spatial calibration is performed (no temporal calibration)

The experiments confirmed that the time offset estimation is crucial as the average metric error of transformation of 3.07 cm and $0.83°$ was measured when the time offset was equal to 40 ms and only increased when the time offset was larger. Similar errors were observed for all analyzed values of standard deviation σ_{LiDAR}. This proves that the users should choose spatiotemporal calibration over spatial

calibration in almost all scenarios (with an exception when hardware synchronization is available).

E. Influence of the number of constraints

In this experiment, the idea was to verify the accuracy of the obtained results based on the number of LiDAR's point-to-plane constraints included in the optimization (Fig. 7).
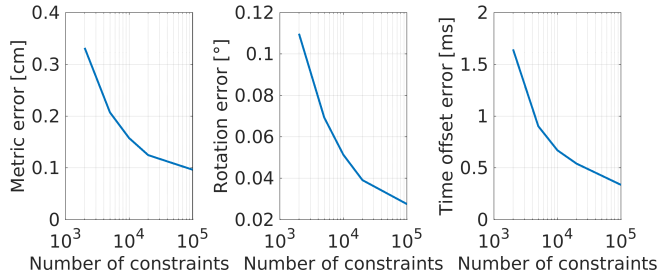

Fig. 7: Metric, rotational and time offset errors of the calibration depending on the number of constraints included in the optimization with $\sigma_{\text{LiDAR}} = 0.01$ m

Results indicate that including more constraints (that are different than those already added) reduces the overall metric, rotation, and time offset errors. Larger number of constraints increases also the optimization time, but still makes it possible to achieve a solution in several minutes in worst case scenarios. What is important, the differences in accuracy between a version with almost all possible constraints (100000) and taking into consideration ten times fewer constraints (10000) is relatively small. From these experimental results we believe that calibration of around 50 seconds, with diverse calibration marker poses in all feasible directions, and rather dynamic motion should be sufficient to obtain accurate calibration under real-world conditions.

F. Influence of the camera framerate

The amount of data used in the optimization depends on the framerate of the camera and 3D LiDAR. In the case of LiDARs, most of the available sensors provide data with 10 Hz but cameras provide possibilities to acquire images even up to hundreds fps. Therefore, we verified the accuracy depending on the chosen framerate from the camera (Fig. 8).
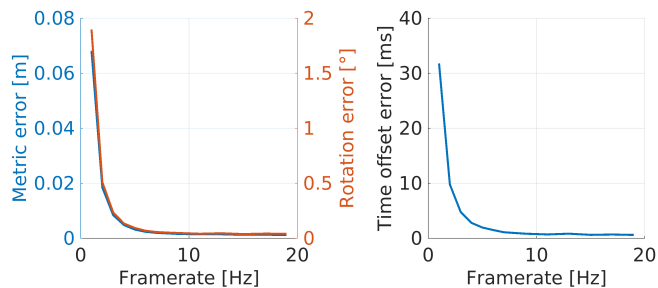

Fig. 8: Metric, rotational and time offset errors of the calibration depending on the framerate of camera chessboard detections with $\sigma_{\text{LiDAR}} = 0.01$ m

Results showed that there is no visible benefit when the camera framerate exceeds 10 Hz. A lower framerate might

negatively influence the accuracy of metric, rotation, and time offset estimations, but higher framerates do not improve the accuracy of the calibration. This proves that the chosen B-spline interpolation works well as long as the interpolation interval does not exceed 100 ms.

V. REAL-DATA EXPERIMENTS

A. Velodyne VLP-16 and stereo-camera setup

The experimental setup for real-world verification comprised of two MV BlueFox3-2016C-1112 global-shutter cameras with Basler Lens C125-0418-5M F1.8 f4mm rigidly mounted on both sides of the Velodyne VLP-16 as presented in Fig. 9. The setup lacks hardware trigger, so, the cameras were software triggered with our custom driver at the same time ensuring close (but not perfect) trigger times.

Fig. 9: The experimental setup with two MV BlueFox3-2016C-1112 cameras and Velodyne VLP-16 LiDAR without hardware trigger (A) that was used for indoor mapping when mounted on a helmet (B)

At first, we calibrated both cameras using *kalibr* [12], [14] achieving a transformation from the left camera to the right camera, $^{\text{right}}\mathbf{K}_{\text{left}}$, that was almost identical to manually measured offset with *kalibr* reporting accuracy of ± 0.06 cm and $\pm 0.0991°$ and thus we treat it as ground-truth reference. Then, we used our software twice to calibrate Velodyne VLP-16 with the left camera ($^{\text{left}}\mathbf{T}_{\text{velo}}$), and to calibrate Velodyne VLP-16 with the right camera ($^{\text{right}}\mathbf{T}_{\text{velo}}$). We compute the difference between both calibrations as:

$$\Delta\mathbf{T} = {}^{\text{right}}\mathbf{K}_{\text{left}} {}^{\text{left}}\mathbf{T}_{\text{velo}} {}^{\text{right}}\mathbf{T}_{\text{velo}}^{-1}. \quad (11)$$

With the original baseline of 31 cm, we measured the translational difference of $\mathbf{T}$ to be 0.74 cm and the rotational difference to be equal to $0.97°$. We obtained time offsets of $\Delta t_1 = 3.99$ ms and $\Delta t_2 = 3.77$ ms for both camera-LiDAR calibrations. The difference between Δt_1 and Δt_2 can be attributed to the imperfect software trigger of our sensory setup solution.

B. MRS6124 and stereo-camera setup

A similar experiment was repeated with a different setup consisting of the same cameras and SICK MRS6124 (Fig. 10), which is a more challenging scenario due to the lower accuracy of SICK MRS6124 (systematic error of ± 12.5 cm and statistical error of 3 cm) when compared to the Velodyne VLP-16. Similarly, we calibrated both cameras using *kalibr* that reported an accuracy of ± 0.0666 cm and $\pm 0.1110°$, and then verified achieved results with two independent

camera-LiDAR calibrations using the proposed software. We measured the difference between *kalibr* and our software to be equal to 0.73 cm and 1.45° for a baseline of approximately 34 cm.

Fig. 10: The experimental setup with two MV BlueFox3-2016C-1112 cameras and SICK MRS6124 LiDAR without hardware trigger (A) that is used in automotive applications (B)

Our calibration indicated that the measured timestamps have to be corrected by 200.99 ms and 202.41 ms for the left and right camera, respectively. In our opinion, such a significant time offset stems from the buffering of data either in the sensor itself or in the ROS driver. The visualization confirms that LiDAR points are significantly delayed when compared to camera images (as presented in Fig. 11). When no temporal calibration would be performed (time offset parameter fixed to 0 in the optimization), we achieved a difference to *kalibr* of 7.51 cm and 4.97°. This experiment confirms the usefulness of our contribution as the time offset was coherently estimated for both camera-LiDAR calibrations and seems correct when visually inspected.
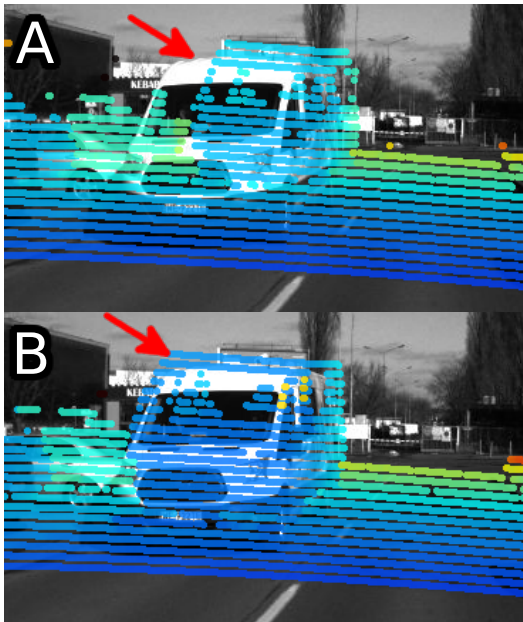

Fig. 11: Visual verification of the calibration of the sensory setup with MRS6124 in the cases of spatial (A) and spatiotemporal (B) calibration. Notice the difference for a moving car marked by red arrows

C. Comparison with state of the art

Fair comparison of spatial or spatiotemporal calibration solutions is challenging as each method has its own, different requirements, and the best performance can be achieved in certain conditions. The accuracy of the calibration strongly depends on the accuracy of the used sensors and the amount of data it provides (in practice the number of scan lines and the angular field of view). Moreover, obtaining precise enough ground-truth for the 7 DoF spatial and temporal relations in order to assess the calibration accuracy is challenging. Nevertheless, we decided to perform extensive evaluation of the accuracy of the proposed solution and then provide an overall comparison with the existing state-of-the-art solutions.

The accuracy of our solution was measured for the stereo-camera setups with Velodyne VLP-16 and SICK MRS6124. Similarly to results reported in previous section we assumed left-right camera calibration from *kalibr* as the ground-truth and compared our results to that result. We run our calibration software 50 times on real data by randomly selecting 35% of all available constraints for both left camera - LiDAR and right camera - LiDAR calibrations achieving 2500 different calibrations to compare to results reported by *kalibr*. We obtained a translation error $t_e = 0.43$ cm and rotation error $r_e = 0.33°$.

The results obtained by our system for the Velodyne sensor show that the proposed system is accurate. The obtained accuracy is comparable to the best reported results by Park [13] but as both reported results used different sources of ground truth (Park used manual ground truth) we would abstain from claiming the best accuracy. From our experiments we also computed the standard deviation of the calibration, that was equal to $\sigma_t = 0.03$ cm and $\sigma_r = 0.005°$ for translational and rotational parts, respectively.

When it comes to SICK MRS6124, we performed the same verification procedure, however, the results cannot be directly compared to the results from [13] and other works using Velodyne LiDARs, due to significantly different measurement characteristics of the SICK LiDAR. Nevertheless, the MRS6124 results demonstrate that the proposed calibration framework is feasible for very different types of LiDARs. Calibrating the SICK sensor we obtained similar translational error $t_e = 0.8$ cm with significantly greater rotational error $t_r = 3.52°$. This is the result of two factors: firstly, MRS6124 has a significantly worse accuracy than Velodyne, and secondly, the laser beams are more tightly packed. Both of these factors combined make it difficult to properly constrain the rotational part of the calibration. We believe that using a larger calibration pattern and recording calibration data with more extreme orientations of this pattern should be sufficient to overcome this issue.

VI. CONCLUSIONS

We propose a novel camera-LiDAR calibration software that is the first marker-based solution that provides spatiotemporal calibration owing to the novel B-spline interpolation of plane equations employing a minimal plane representation in

Lie algebra. The solution requires only a commonly available calibration marker and a short, one-minute calibration session to provide repeatable and accurate results. Moreover, it is available as a ROS-compatible package.

The simulation experiments over a significant number of random scenarios show that we should expect a centimeter-level accuracy when using currently available 3D LiDARs, while demonstrating also, that our solution is robust to bad spatial and temporal initial guesses. We performed two real-world experiments with Velodyne VLP-16 and SICK MRS6124 LiDARs. The indirect comparison with *kalibr* shows that our method provides results of similar accuracy. This suggests that the presented calibration procedure can be practically combined with *kalibr* in the ROS environment whenever calibration of a more complicated sensory system with a 3D LiDAR and a camera is necessary. The experimental results demonstrate also that the temporal calibration is critical for applications and should be performed when no hardware synchronization is available.

REFERENCES

[1] K. Huang, J. Xiao, and C. Stachniss, "Accurate direct visual-laser odometry with explicit occlusion handling and plane detection", 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, 1295–1301.

[2] J. A. Castellanos, J. Neira, and J. D. Tardos, "Multisensor fusion for simultaneous localization and map building", in IEEE Transactions on Robotics and Automation, 17(6), 2001, 908–914.

[3] Q. Zhang, and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)", 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, 2004, 2301–2306.

[4] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes", 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, 2007, 4164–4169.

[5] M. Velas, M. Spanel, Z. Materna, and A. Herout, "Calibration of RGB camera with velodyne LiDAR", 2014 WSCG Communication Papers Proceedings, Plzen, 2014, 135–144.

[6] X. Gong, Y. Lin, and J. Liu, "3D LIDAR-camera extrinsic calibration using an arbitrary trihedron", MDPI Sensors, 2013, vol. 13, no. 2, 1902–1918.

[7] Z. Pusztai, and L. Hajder, "Accurate calibration of LiDAR-camera systems using ordinary boxes", 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, 2017, 394–402.

[8] A. Fornaser, P. Tomasin, M. De Cecco, M. Tavernini, and M. Zanetti, "Automatic graph based spatiotemporal extrinsic calibration of multiple Kinect V2 ToF cameras", in Robotics and Autonomous Systems, 98, 2017, 105–125.

[9] R. Ishikawa, T. Oishi, and K. Ikeuchi, "LiDAR and camera calibration using motions estimated by sensor fusion odometry", 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, 7342–7349.

[10] Z. Taylor, and J. Nieto, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation", in IEEE Transactions on Robotics, vol. 32, no. 5, 2016, 1215–1229.

[11] J. Peršic, L. Petrović, I. Marković, and I. Petrović, "Spatio-temporal multisensor calibration based on gaussian processes moving object tracking", arXiv preprint, 1904.04187v1, 2019.

[12] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems", 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, 1280–1286.

[13] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-LiDAR calibration: a targetless and structureless approach", in IEEE Robotics and Automation Letters, vol. 5, no. 2, 2020, 1556–1563.

[14] J. Rehder, R. Siegwart, and P. Furgale, "A general approach to spatiotemporal calibration in multisensor systems", in IEEE Transactions on Robotics, vol. 32, no. 2, 2016, 383–398.

[15] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka, "Optimization based IMU camera calibration", 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, 2011, 3297–3304.

[16] J. Zhang, and S. Singh, "LOAM: Lidar odometry and mapping in real-time", Robotics: Science and Systems Conference (RSS), Pittsburgh, PA, 2014.

[17] S. Lovegrove, A. Patron-Perez, G. Sibley, "Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras", British Machine Vision Conference (BMVC), 2013.

[18] M. Bosse, R. Zlot, and P. Flick, "Zebedee, Design of a spring-mounted 3D range sensor with application to mobile mapping", IEEE Transactions on Robotics, vol. 28, no. 5, 2012, 1104–1119.

[19] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization", IEEE International Conference on Robotics and Automation, Shanghai, 2011, 3607–3613.

[20] A. Bartoli, "On the non-linear optimization of projective motion using minimal parameters" European Conference on Computer Vision (ECCV), Copenhagen, 2002, 340–354.

[21] S. R. Buss and J. P. Fillmore, "Spherical averages and applications to spherical splines and interpolation", ACM Transactions on Graphics, vol. 20, no. 2, 2001, 95–126.

[22] J. Deray and J. Solà, "Manif: A micro Lie theory library for state estimation in robotics applications", Journal of Open Source Software, vol. 5, no. 46, 2020, 1371.

[23] M. J. Kim, M. S. Kim, S. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives", in ACM Special Interest Group on Computer GRAPHics and Interactive Techniques Conference (SIGGRAPH), Los Angeles, 1995, 369–376.

[24] A. Patron-Perez, S. Lovegrove, and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras", in International Journal of Computer Vision, vol. 113, no. 3, 208–219, 2015.

[25] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative B-splines on Lie groups", arXiv preprint, 1911.08860v1, 2019.

[26] P. Skrzypczyński, "Spatial uncertainty management for simultaneous localization and mapping", 2007 IEEE International Conference on Robotics and Automation (ICRA), Rome, 2007, 4050–4055.