MOISST: Multimodal Optimization of Implicit Scene for SpatioTemporal calibration

Quentin Herau[1,2]   Nathan Piasco[1]   Moussab Bennehar[1]   Luis Roldão[1]   Dzmitry Tsishkou[1]
Cyrille Migniot[3]   Pascal Vasseur[4]   Cédric Demonceaux[2]

[1]Noah's Ark, Huawei Paris Research Center   [2]ICB UMR CNRS 6303, Université de Bourgogne
[3]ImViA UR 7535, Université de Bourgogne   [4]MIS UR 4290, Université de Picardie Jules Verne

`{Quentin.Herau, Nathan.Piasco, Moussab.Bennehar, Luis.Roldao, Dzmitry.Tsishkou}@huawei.com`

`{Quentin.Herau@etu, Cyrille.Migniot@, Cedric.Demonceaux@ }@u-bourgogne.fr`

`Pascal.Vasseur@u-picardie.fr`

Abstract

With the recent advances in autonomous driving and the decreasing cost of LiDARs, the use of multimodal sensor systems is on the rise. However, in order to make use of the information provided by a variety of complimentary sensors, it is necessary to accurately calibrate them. We take advantage of recent advances in computer graphics and implicit volumetric scene representation to tackle the problem of multi-sensor spatial and temporal calibration. Thanks to a new formulation of the Neural Radiance Field (NeRF) optimization, we are able to jointly optimize calibration parameters along with scene representation based on radiometric and geometric measurements. Our method enables accurate and robust calibration from data captured in uncontrolled and unstructured urban environments, making our solution more scalable than existing calibration solutions. We demonstrate the accuracy and robustness of our method in urban scenes typically encountered in autonomous driving scenarios.

1. Introduction

Most robotic and intelligent systems rely heavily on sensory information to achieve various tasks. Moreover, commonly encountered sensor setups for autonomous driving consist of multiple sensors acquiring different data modalities (e.g. cameras, LiDARs, IMUs, GNSS systems, etc.) which can greatly improve the performance on different tasks such as mapping [2], localization [28] and perception [33].

However, to correctly exploit and merge the information provided by all sensors, it is important to represent their data in a common reference frame. Spatial extrinsic calibration is the process that determines the relative geometric
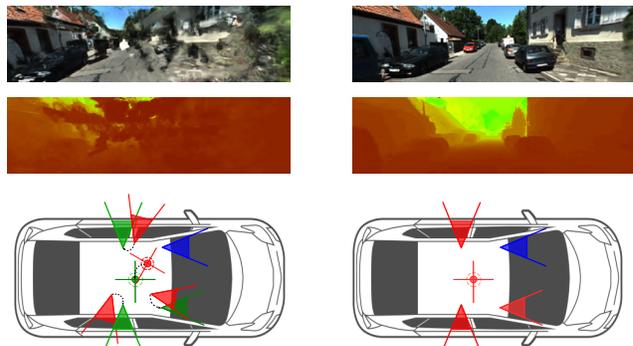

Figure 1. Effect of calibration on novel view synthesis: training positions in red, ground truth positions in green, reference position in blue. The RGB images (top) and depth maps (middle) are rendered from an implicit neural 3D scene trained from non-calibrated (left) and calibrated with MOISST (right) sensors.

transformation between the sensor poses by considering a 6-DoF rigid-body transformation. Although accurate spatial calibration is essential in multi-sensor setups, it is often not sufficient due to time synchronization issues between the different sensors. Time synchronization is the process that determines the time offset between the different sensor measurements, in the case there is no hardware synchronization which would eliminate any delay.

Current existing methods commonly require the use of calibration targets placed in the scene to fuse all sensors in a common frame [35, 3]. This is unpractical in many cases, especially for dynamic tasks, where the calibration setup must be redone regularly. Although some papers offer solutions to bypass this constraint by detecting salient geometric features (e.g. edges [15, 34], planes [18]) within acquired scenes, such features might not be present in all kind of environments.

Furthermore, these methods often do not consider the possible asynchronization of the sensors. The effect of a wrongly synchronized rig can have a consequential impact on the performance depending on the task. Moreover, unconsidered time offsets within calibration can strongly degenerate extrinsic estimation leading to suboptimal results.

Considering all the issues mentioned above, we introduce our new method called MOISST: Multimodal Optimization of Implicit Scene for SpatioTemporal calibration. MOISST is a novel calibration method which leverages an implicit neural 3D scene representation known as Neural Radiance Fields (NeRF) [11]. It can be trained with any kind of sensor providing radiometric or geometric information on a given 3D scene. This representation is by nature the common reference frame used for the sensor fusion. We take advantage of the differentiable property of our scene representation to simultaneously learn the scene's geometry and colors, and the poses given to the neural network. Unlike existing NeRF-based methods of pose regression [32, 6, 9], we consider the rigid constraint in the multi-sensor rig to reduce the number of optimized parameters. By using a time-parameterized differentiable formulation for the main sensor trajectory, we can also detect and compensate potential time offsets between the sensors. To the best of our knowledge, tackling the problem of multi-sensor spatiotemporal calibration using implicit representations has not been proposed before in the literature.

Thanks to our formulation, we are able to propose an offline targetless solution to spatiotemporal calibration of multimodal sensors, that is also structureless, as we do not require specific geometric structures like edges or planes in the scene for our method. Compared to other methods and because of the aforementioned characteristics of our solution, MOISST is especially adapted to perform automatic re-calibration of a multi-sensor device during the full lifecycle of the system. MOISST is a simple – *it can be run from acquisition data recorded in any environment* – and inexpensive – *it does not require target or external hardware* – calibration solution, which are crucial features for robots and large scale fleet of autonomous vehicles.

2. Related works

2.1. Multimodal extrinsic and temporal calibration

Extrinsic calibration for multimodal sensors is a well studied subject that can be categorized in two main groups: target-based and targetless methods.

2.1.1 Target-based calibration

Zhang *et al.* [35] were the first to introduce the use of a planar checkerboard target for camera and laser range finder calibration, by using the latter to determine the checker-

board plane, and the pattern seen by the camera to calculate its pose. Three pairs of capture measurements are enough to deduce extrinsic parameters between both sensors. Geiger *et al.* [3] propose a solution to calibrate the sensors with a single capture, by placing multiple targets in the scene. While these methods provide satisfactory calibration accuracy, they necessitate the placement of targets in the scene, which might not be available or practical to place in typical real-world scenarios.

2.1.2 Targetless calibration

Targetless methods usually use the concept of mutual information, by matching corresponding elements obtained through different type of sensors. This can be edges for visible cameras, depth gradient for LiDARs [15, 34], or the use of correspondence between image intensity and surface normals [26]. However, by relying on specific geometric features, these methods only work in a well structured scene with noticeable, recognizable and detectable patterns such as straight lines or edges and often work only in indoor environments. There is also a set of deep learning based methods [20, 5, 10], able to find the transformation between a camera capture and a LiDAR scan. However, these methods are trained in a supervised manner, needing a labeled dataset, and are prone to overfitting, limiting their use to environments reflecting the training dataset.

Although the previously mentioned methods achieve satisfactory performance given ideal conditions, they suppose a perfectly time-synchronized set of sensors, which is possible through specific hardware [23] but is often challenging and sensor dependant (i.e. most low-cost cameras do not support such features). There exists some methods tackling temporal calibration, both target-based [7], or targetless [18], they require however additional sensors such as calibrated camera-IMU pair to obtain a precise trajectory. With the methods proposed by Taylor *et al.* [27] and Park *et al.* [17], visual and LiDAR odometry are used for calculating trajectories for each sensor and matching them, allowing both spatial and temporal calibration. Nevertheless, this approach can generate a progressive drift with the accumulated transformations between the frames.

Contrary to the previously mentioned methods, MOISST does not require any targets (i.e. targetless) and determines both spatial and temporal calibration parameters by only relying on the poses of a single sensor, avoiding the cumulative errors in the different per-sensor trajectories. At the same time, we fuse information from all sensors, and thanks to the use of a dense implicit scene representation, we do not require specific geometric structure (i.e. structureless). This approach allows compatibility with a greater variety of scenes and allows us to scale to almost all real-world scenarios.

2.2. Neural 3D scene representation

NeRF [11] is an implicit representation of a 3D scene primarily used for novel view synthesis. From a set of images and their corresponding camera poses, the model learns the 3D geometry through differentiable volume rendering. NeRF provides a continuous representation, resulting in improved rendering fidelity and compactness compared to classical explicit scene representations [29]. Beyond the rendering ability, many recent methods have used these implicit scene representations for downstream robotics tasks [24, 4, 12].

NeRF stores all the color and density information of the scene in a multilayer perceptron (MLP), and allows any rendering resolution as the representation is continuous. The model takes as input a 3D coordinate and a direction vector, outputs a color and density information for this 3D point, and is trained through a differentiable rendering procedure. A sinusoidal encoding [25] of the input coordinate maps the low dimensional 3D position and direction to a higher dimension representation, allowing the rendering of a highly detailed scene representation. To speed-up convergence, Instant Neural Graphics Primitives (Instant-NGP) [13] was introduced, allowing much faster convergence with higher quality rendering. It uses a multi-resolution hash encoding instead of sinusoidal encoding, considerably reducing the size of the trained MLP.

The training of NeRF requires mainly RGB images from cameras, optionally depth information such as point clouds from LiDARs [19], along with registered poses for each sensor frame. The final rendering quality is highly dependant on the precision of these poses, as seen in Fig. 1, where the result without optimization has incorrect geometry, produces low quality novel views and is not usable. As an answer to this limitation, NeRF^{--} [32] exploits the fully differentiable structure of NeRF to not only train the NeRF model, but also to optimize the camera poses. This makes the model robust to noisy poses, as it is able to optimize both the camera poses alongside the NeRF model. SCNeRF [6] uses the first two columns of the rotation matrix to formulate rotations instead of Rodrigues formula to achieve better convergence, and BARF [9] improves upon these methods by using low-to-high frequency release for the input positional encoding, avoiding local minima during pose optimization.

In our proposal, because we focus on multi-sensor calibration, we only optimize the extrinsic transformation between the sensors instead of optimizing each pose of each frame independently. Indeed, in a rigid sensor setup, sensors are not allowed to move freely relative to each other. Compared to aforementioned methods, this novel and calibration-focused formulation reduces the number of parameters to be optimized and is more robust to outliers thanks to the rigidity constraint imposed on each pose. The proposed formulation also allow us to optimize the time offset between sensors, which is often hard to be achieved within the same optimization framework and may require additional information.

3. Method

3.1. Notations and background

3.1.1 Notations

We consider a multi-sensor system with S sensors with $r \in [1, S]$ being our reference sensor and each sensor is either a camera or a LiDAR. We use the following notations to describe our method:

- $\{N_i\}_{i \in [1,S]}$: set of number of frames captured by each sensor,

- $n_i, n \in [1, N_i]$: index of frames captured by sensor i,

- $t^{n_i} \in \mathbf{R}^+$: the absolute timestamp of frame n_i relative to the sensor i clock,

- $\delta_i \in \mathbf{R}$: the time offset between the reference frame clock and the sensor i clock,

- $_wT^i(t) \in \mathbf{R}^{4 \times 4}$: the pose transformation matrix of sensor i at time t (time relative to sensor i clock) in the world reference,

- $_jT^i \in \mathbf{R}^{4 \times 4}$: the extrinsic homogeneous transformation matrix from sensor i to sensor j.

We aim to calibrate our system according to the reference sensor. The goal is to obtain the transformation matrices $_jT^i$ and time offsets δ_i between the sensors to calibrate and the reference one. We consider that we know the pose of sensor r in a global frame, which could be easily obtained through SLAM [14] or Structure-from-Motion [21]. We also consider its clock as the reference clock, $\delta_r = 0$. From poses of reference sensor, $_wT^r(t^{n_r}), n_r \in [1, N_r]$, we build a continuous trajectory. We do that by interpolating between the existing poses, and extrapolating outside the defined temporal bounds by extending the transformations at the beginning and the end of the sequence. This modeling process is very similar to what is defined in [17]. For the interpolation functions, we used spherical linear interpolation (SLERP) [22] for the rotation, and linear interpolation (LERP) for the translation. We denote this interpolation function as $\mathcal{T}_r$:

$$_wT^r(t) = \mathcal{T}_r(t). \tag{1}$$

3.1.2 Spatiotemporal calibration

Given the spatial extrinsic calibration and the time offset of the other sensors regarding our reference sensor, we can
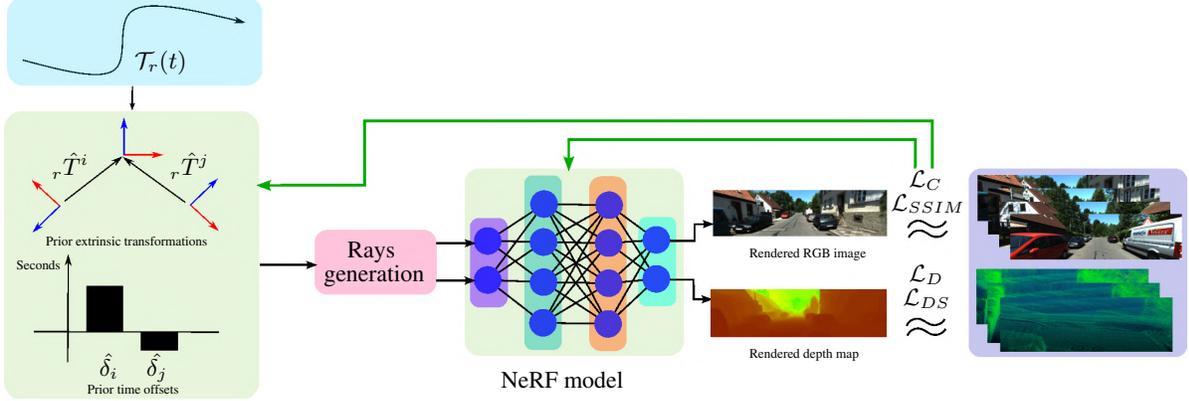
Figure 2. Overview of MOISST optimization framework. First, the model is initialized with rays generated using rough spatial and temporal calibration priors in addition to the reference frame trajectory. After each optimization step, the rays are regenerated and fed to the NeRF model. We then render RGB images and depth maps which are used along the ground truth ones to compute the losses and propagate the gradients. Gradient descent algorithm is finally used to optimize both NeRF and calibration parameters.

compute the pose of sensor i in a global frame with the formula:

$$_wT^i(t^{n_i} + \delta_i) = \mathcal{T}_r(t^{n_i} + \delta_i) \, _iT^r. \tag{2}$$

3.1.3 Implicit neural scene representation

An implicit neural representation models a scene with a neural network by mapping coordinates as inputs to quantities of interests, such as color or density, as outputs. By evaluating points along camera rays and composing their densities and colors through volumetric rendering, such methods can synthesize RGB images and depth maps[1] from an arbitrary sensor pose.

In order to train said neural network on a specific scene, it is necessary to have a training set with sensor information and a pose associated. This information may be an RGB image in the case of visible camera, or a point cloud in the case of a LiDAR. We aim to find the parameters of the neural network Θ that minimizes the difference between the provided information (I^{n_i} - image n_i of sensor i - or D^{n_i} - depth information n_i of sensor i) and the rendered result by the model defined as:

$$\mathcal{R}_I \left(_wT^i(t^{n_i}) \mid \Theta \right), \tag{3}$$
$$\mathcal{R}_D \left(_wT^i(t^{n_i}) \mid \Theta \right), \tag{4}$$

with $\mathcal{R}_I$ being the model inference and ray composition function that returns a RGB image prediction of frame n_i for sensor i and $\mathcal{R}_D$ being equivalent to $\mathcal{R}_I$ but returning rays depth instead of colors. By minimising the loss $\mathcal{L}_{total}$

---

[1]We can estimate depth of ray with alpha composition of distances from the center of the ray to the sampled points.

defined as:

$$\mathcal{L}_{total} = \lambda_C \mathcal{L}_C + \lambda_D \mathcal{L}_D, \tag{5}$$

$$\mathcal{L}_C = \sum_{i=1}^{S} \sum_{n_i=1}^{N_i} \left\| \mathcal{R}_I \left(_wT^i(t^{n_i}) \mid \Theta \right) - I^{n_i} \right\|_2^2, \tag{6}$$

$$\mathcal{L}_D = \sum_{i=1}^{S} \sum_{n_i=1}^{N_i} \left\| \mathcal{R}_D \left(_wT^i(t^{n_i}) \mid \Theta \right) - D^{n_i} \right\|_2^2, \tag{7}$$

with λ_C, λ_D weighting hyper-parameters, we can estimate the optimal network parameters $\hat{\Theta}$ satisfying:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}(\mathcal{L}_{total}). \tag{8}$$

As explained by Wang *et al.* [32], because the scene representation we use is fully differentiable, it is possible to optimize the input poses with gradient descent jointly with the radiance field parameters. The optimization objective becomes the following:

$$\left\{ \hat{\Theta}, _w\hat{T}^i \right\} = \underset{\Theta, _wT^i}{\operatorname{argmin}}(\mathcal{L}_{total}). \tag{9}$$

3.2. MOISST Optimization formulation

In this section, we introduce our novel optimization formulation for multi-sensor system spatiotemporal calibration. Considering a multi-sensor system such as a robot or an autonomous car, we know that the poses of each sensor observation are not independent, as there exists a rigid transformation between each sensor. Because we know the trajectory $\mathcal{T}_r$ of the reference sensor r, we can express the absolute pose of each remaining sensor according to sensor r (see equation 2). Substituting $_wT^i$ in equation 3 leads to the following formulation:

$$\mathcal{R}_I \left(\mathcal{T}_r(t^{n_i} + \delta_i) \, _iT^r \mid \Theta \right). \tag{10}$$

Similar reasoning can be made for depth rendering function $\mathcal{R}_D$. Our new formulation of the rendering functions can be integrated in the color loss of equation 6:

$$\mathcal{L}_C = \sum_{i=1}^{S} \sum_{n_i=1}^{N_i} \left\| \mathcal{R}_I \left(\mathcal{T}_r(t^{n_i} + \delta_i) \, _iT^r \mid \Theta \right) - I^{n_i} \right\|_2^2.$$
(11)

We can replace the depth rendering function in the same manner in equation 7. This leads to our new optimization formulation:

$$\left\{ \hat{\Theta}, \, _i\hat{T}^r, \hat{\delta}_i \right\} = \underset{\Theta, \, _iT^r, \delta_i}{\mathrm{argmin}} \, (\mathcal{L}_{total}),$$
(12)

with $_i\hat{T}^r$ and $\hat{\delta}_i$ the only parameters to optimize along with the network weights. As the trajectory $\mathcal{T}_r$ is continuous over time, we can also optimize the time offsets δ_i. With the proposed method, we only have to optimize the extrinsic transformation between all sensors and the reference sensor, reducing the number of optimized parameters compared to the full set of frame poses as in equation 9. A summary of our proposal is shown in Fig. 2.

3.3. Optimization details

3.3.1 Additional losses for geometric consistency

We add two more losses using image patches to further improve the geometry of the NeRF model and the proper estimation of calibration parameters. The first loss is the structural dissimilarity (DSSIM) $\mathcal{L}_{SSIM}$ [31], which minimizes the difference in local 2D structures between the rendered and the input image [30]. The second is the depth smoothness loss $\mathcal{L}_{DS}$, also used by RegNeRF [16], which regularizes the depth variation in randomly selected patches of the images to reduce variation of the predicted depth. Our final loss function becomes as follows:

$$\mathcal{L}_{total} = \lambda_C \mathcal{L}_C + \lambda_D \mathcal{L}_D + \lambda_{SSIM} \mathcal{L}_{SSIM} + \lambda_{DS} \mathcal{L}_{DS}$$
(13)

with $\lambda_C, \lambda_D, \lambda_{SSIM}, \lambda_{DS}$ the weight factor for each loss.

3.3.2 Network architecture

We use an implicit scene representation similar to the `nerfacto` model of Nerfstudio[2] open source framework. It is inspired by the proposal network introduced in MipNeRF 360 [1] with two proposal radiance fields and one final radiance field that outputs the color and density for the volumetric rendering. The proposals are used to samples points along the rays where the density is high. We use the hash

grid introduced in instant NGP [13] for positional encoding and spherical harmonics for directional encoding. We found this model to be a good trade-off between speed and accuracy for our spatiotemporal calibration problem.

3.3.3 Regularization

In BARF [9], the idea of low-to-high frequency release for the positional encoding allows smoothness in the scene, which helps the optimization of the poses to avoid local minima. With our architecture using the instant-NGP backbone, we do not have a sinusoidal positional encoding, but a multi-resolution hash grid instead. In order to mimic BARF, we introduce a weight decay to the hash encoder for a few epochs, before removing it, allowing higher frequency information to be learned afterwards. We also wait a few epochs before applying the depth loss L_D, as we found that initializing the geometry solely through visual supervision helps the whole system to converge better.

3.3.4 Spatiotemporal priors

We use spatial calibration priors to initialize the extrinsic parameters of the sensors. For the temporal shift, we set the initial estimate to 0 as we found that our solution was very robust to temporal calibration. Ablation on the sensibility of our method to initial prior is provided in section 4.5.

4. Experiments

We evaluate MOISST on the NVS training set from the recent KITTI-360 dataset [8] which involves difficult static outdoor scenarios with 2 forward and 2 side facing cameras and a top-mounted Velodyne HDL-64E LiDAR sensor. The Lidar scans provided by the KITTI-360 dataset are undistorted using motion compensation. We report results on sequences 0, 1, 2 and 4 and consider the front-left camera as our reference sensor r for all experiments[3]. We apply ± 50 cm translation and $\pm 5°$ rotation offsets on all axes and ± 100 ms time offset to simulate spatial and temporal calibration errors, respectively. The geodesic distance and the classical L_2-norm are used to report the rotation and translation errors along all axes, and the average error over the last 10 epochs is reported for fairness. Each training takes between 1 and 4 hours depending on the number of selected sensors on a high-end consumer grade GPU.

Implementation details: Given the sparser supervision signal from re-projected LiDAR depth maps, color and depth losses are balanced by $\lambda_C = 1$ and $\lambda_D = 20$. Furthermore, as radiance fields require initial optimization to

---

[2]https://docs.nerf.studio/en/latest/nerfology/methods/nerfacto.html

[3]Experiments are not performed on Sequence 3 as it has missing captures of LiDAR scans.

Table 1. Spatial calibration accuracy.

| | Front-right camera | | LiDAR | |
Sequence	Translation error (cm)	Rotation error (°)	Translation error (cm)	Rotation error (°)
0	2.3 ± 0.6	0.09 ± 0.02	8.2 ± 1.8	0.21 ± 0.08
1	1.6 ± 0.4	0.09 ± 0.02	7.3 ± 2.1	0.41 ± 0.08
2	1.2 ± 0.6	0.04 ± 0.03	13.5 ± 1.8	0.18 ± 0.10
4	2.2 ± 0.7	0.08 ± 0.06	9.9 ± 2.6	0.30 ± 0.10

learn depth through color supervision, depth loss is applied after two epochs. Geometric consistency losses are empirically balanced by $\lambda_{SSIM} = 0.1$ and $\lambda_{DS} = 0.0001$. We use the Adam optimizer and train our network during 50 epochs for all experiments. We start with a learning rate of 1×10^{-2} for the network parameters and 5×10^{-5} for the spatial and temporal parameters, and exponentially decay to a factor of 1×10^{-2} of the original learning rate. We apply a weight decay of 1×10^{-6} on the network parameters, including the hash grid, then remove the weight decay of the hash grid after 5 epochs as explained in section 3.3.3.

We perform extensive evaluation of our solution upon three different conditions: in section 4.1 with a scenario where we consider only spatial extrinsic noise, in section 4.2 with only temporal miscalibration and finally in section 4.3 by taking into account both spatial and temporal calibration parameters.

4.1. Spatial Calibration

In this section, we only consider a spatial error, and remove the time offsets. We only optimize spatial parameters. As we can see in Table 1, MOISST estimates calibration parameters with low rotation error, below 1° in all sequences. As for translation, we are able to reach around 2 cm of error on the front-right camera, and between 7 cm and 14 cm for the LiDAR. The higher error in LiDAR translation calibration needs to be mitigated because of the relative precision of the provided ground truth, as explained in section 4.4.2.

4.2. Temporal Calibration

In this section, we only consider a temporal error, and remove the initial spatial error. We only optimize temporal parameters. As shown by the results in Table 2, our method

Table 2. Temporal calibration accuracy.

Sequence	Initial error (ms)	Front-right camera Temporal error (ms)	LiDAR Temporal error (ms)
0	100	0.7 ± 0.3	7.6 ± 1.0
	200	0.5 ± 0.3	6.2 ± 2.0
1	100	0.2 ± 0.2	1.5 ± 1.1
	200	0.5 ± 0.3	1.1 ± 0.5
2	100	1.0 ± 0.3	14.0 ± 1.8
	200	1.1 ± 0.3	13.8 ± 1.2
4	100	0.6 ± 0.3	3.4 ± 1.5
	200	0.7 ± 0.3	14.2 ± 1.5

in able to temporally calibrate the front-right camera with a precision under 1 ms, despite the high initial time offset of 100 ms or 200 ms, knowing the camera is capturing at around 10 fps. The LiDAR's final temporal error is more variable depending on the scene, reaching around 15 ms at maximum. As mentioned previously, this higher error might partially explain by the relative precision of the provided ground truth (see section 4.4.2).

4.3. Combined Calibration

In this section, we consider the full initial error as described in section 4. We run MOISST on all four cameras and the LiDAR of the KITTI-360 dataset. We can see in Table 3 that our method is able to calibrate the 2 side cameras, looking in completely different directions than our reference sensor. The performance is variable depending on the sequence. For example, the sequence 0 is captured in a very narrow road, giving the side cameras a small FOV, reducing the overlap and causing a drop in accuracy.

4.4. Discussions

4.4.1 Comparison with structure-based methods

We wanted to compare our camera-LiDAR calibration results with the structure-based method from Yuan *et al.* [34], but we could only obtain subpar results with their method on the dataset we use (we obtained a mean translation error of 60 cm and 4.08° of rotational error starting from a translation and rotational error of 50 cm and 5°, respectively, on all axes). We found that it needed a denser point cloud from the LiDAR than what was provided in the KITTI-360 dataset in order to find reliable edge features in the scene. In addition, compared to our solution, this method is not able to do camera/camera and LiDAR/LiDAR calibration, or calibrate temporally.
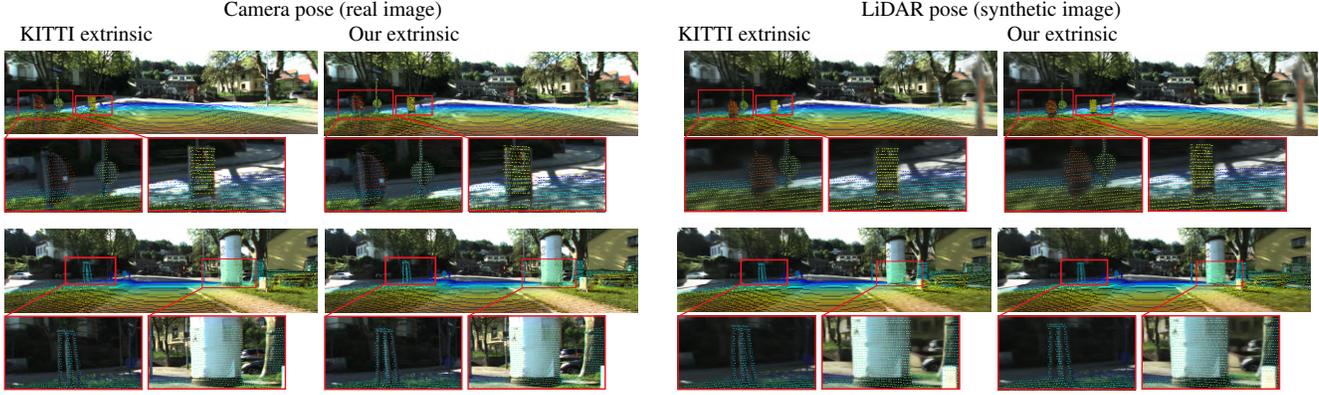
Figure 3. Limitation on KITTI-360 LiDAR ground truth calibration: we compare the alignment of re-projected 3D points from the LiDAR on the front image using KITTI extrinsic calibration and our optimized extrinsic calibration. We also re-project the 3D points on a synthetic image generated with the same pose as the LiDAR on the vehicle, in order to avoid parallax effect.

Table 3. Spatiotemporal calibration accuracy.

	Front-Right Camera			Left-side Camera		
Sequence	Translation error (cm)	Rotation error (°)	Temporal error (ms)	Translation error (cm)	Rotation error (°)	Temporal error (ms)
0	1.1 ± 0.3	0.08 ± 0.02	0.5 ± 0.3	10.0 ± 0.7	0.41 ± 0.24	0.5 ± 0.2
1	2.0 ± 0.3	0.09 ± 0.02	0.9 ± 0.3	1.6 ± 0.6	0.16 ± 0.03	0.5 ± 0.3
2	1.8 ± 1.0	0.09 ± 0.05	0.5 ± 0.4	3.0 ± 0.6	0.06 ± 0.05	2.1 ± 0.3
4	1.9 ± 0.7	0.05 ± 0.01	1.3 ± 0.6	2.7 ± 0.4	0.05 ± 0.25	0.6 ± 0.5
	Right-side Camera			LiDAR		
0	5.2 ± 0.7	0.12 ± 0.02	1.4 ± 0.4	9.3 ± 2.3	0.50 ± 0.17	0.04 ± 0.02
1	8.4 ± 0.7	0.26 ± 0.02	1.5 ± 0.4	10.6 ± 3.0	0.59 ± 0.08	6.9 ± 2.1
2	3.0 ± 0.9	0.07 ± 0.05	1.5 ± 0.6	15.8 ± 2.9	0.23 ± 0.11	2.3 ± 1.7
4	7.1 ± 0.6	0.08 ± 0.03	6.3 ± 0.8	14.0 ± 2.1	0.37 ± 0.09	22.4 ± 1.8

Table 4. Calibration accuracy with solely rotation or translation error.

		Front-Right Camera		LiDAR	
Error type	Initial error	Translation error (cm)	Rotation error (°)	Translation error (cm)	Rotation error (°)
	2°	1.6 ± 0.3	0.09 ± 0.01	9.2 ± 1.9	0.39 ± 0.09
Rotation all axes	5°	1.7 ± 0.5	0.07 ± 0.05	9.9 ± 2.3	0.4 ± 0.14
	10°	127.2 ± 0.6	15.81 ± 0.02	104.4 ± 1.7	17.07 ± 0.13
	20cm	1.8 ± 0.4	0.09 ± 0.02	8.7 ± 1.5	0.46 ± 0.1
Translation all axes	50cm	1.7 ± 0.5	0.09 ± 0.02	7.8 ± 1.2	0.5 ± 0.06
	100cm	1.7 ± 0.3	0.09 ± 0.02	8.8 ± 2.4	0.43 ± 0.08

Table 5. Ablation study on optimized parameters.

	Front-Right Camera			LiDAR		
Optimized parameters	Translation error (cm)	Rotation error (°)	Temporal error (ms)	Translation error (cm)	Rotation error (°)	Temporal error (ms)
Only spatial	103.0 ± 1.0	1.03 ± 0.03	–	104.8 ± 2.3	0.35 ± 0.07	–
Only temporal	–	–	89.6 ± 0.2	–	–	100.9 ± 0.8
Spatial & temporal	1.9 ± 0.6	0.08 ± 0.01	1.2 ± 0.3	11.5 ± 2.1	0.44 ± 0.07	10.1 ± 1.3

Table 6. Poses accuracy for different optimized parameters.

	Front-Right Camera		LiDAR	
Optimized parameters	Translation error (cm)	Rotation error (°)	Translation error (cm)	Rotation error (°)
Only spatial	9.2 ± 1.2	0.44 ± 0.09	12.6 ± 1.7	0.54 ± 0.08
Only temporal	74.8 ± 0.8	8.8 ± 0.05	90.4 ± 1.1	8.8 ± 0.06
Spatial & temporal	2.0 ± 0.4	0.09 ± 0.02	8.9 ± 3.2	0.42 ± 0.09

4.4.2 Limitation of KITTI-360 LiDAR ground truth calibration

In our experiments, we found that the extrinsic calibration between the front camera and the LiDAR provided by

KITTI-360 might be accurate only up to a few centimeters. To show this, we performed the following experiment: we re-projected the LiDAR points into the images captured by the front camera according to: 1) the provided ground truth calibration, 2) the extrinsic calibration we obtained after optimizing the spatiotemporal parameters. We also provided alignment comparison using NeRF generated images at the same location as the LiDAR position on the vehicle to avoid parallax effect. The LiDAR is positioned on top of the camera: some points re-projected on the images should not be visible from the camera position. Results are presented in Fig. 3 (more results in the supplementary video). Comparing the alignment between the re-projected 3D points on the real and synthetic images, we clearly see that our extrinsic calibration seems more accurate than the ground truth we use to compare our results with in this paper.

4.5. Ablation studies

For the ablation studies, we only run our experiments on sequence 1.

4.5.1 Rotation vs Translation error

By running the training with solely rotation or translation errors of varying levels, we could observe that the initial rotation error has more impact on the final accuracy, as we did not get a satisfactory calibration when we introduced $10°$ rotation error on all axis. The results are shown in Table 4. On the contrary, the translation error is well-handled, even with 100 cm error set initially on all axis.

4.5.2 Spatiotemporal coupling

We run an ablation study on the optimized parameters and report the results in Table 5 and Table 6. It shows that if there is spatial and temporal errors and only one of them is optimized, it is not possible to obtain a correct calibration. Which means it is necessary to take into account both type of error. We can observe in Table 6 that optimizing only the spatial parameters allows decent pose errors, showing that they are partly compensating the time offsets. This is possible because sequence 1 is mostly a straight line with the car driving at almost constant speed.

Table 7. Ablation study on losses with spatiotemporal calibration (**best** and <u>second best</u>).

Loss	Translation error (cm)	Rotation error (°)	Temporal error (ms)
$L_C + L_D$	7.4 ± 1.2	0.31 ± 0.06	4.4 ± 1.0
$L_C + L_D$ + SSIM	<u>6.9 ± 1.3</u>	<u>0.30 ± 0.04</u>	**1.9 ± 0.6**
$L_C + L_D$ + DS	14.4 ± 2.1	0.31 ± 0.14	4.4 ± 0.8
$L_C + L_D$ + SSIM+DS	**6.8 ± 1.0**	**0.28 ± 0.06**	<u>2.8 ± 0.6</u>

4.5.3 Ablation on additional losses

In Table 7, we demonstrate that the overall accuracy of our method increases when $\mathcal{L}_{SSIM}$ and $\mathcal{L}_{DS}$ are used. $\mathcal{L}_{SSIM}$ has the largest impact on the performance as it help the implicit scene representation to learn a proper and sharp geometry from radiometric signals. It makes sense that better scene geometry improves the calibration accuracy, especially between LiDARs and cameras.

5. Conclusions and future works

We presented in this paper MOISST, a novel approach based on implicit neural scene representation to spatially and temporally calibrate a multi-sensor system. The proposed approach has the advantage of being scalable to any number of cameras and LiDARs by relying on the trajectory of a single reference sensor. The proposed approach does not require any targets, or specific geometric structure within the scene to achieve accurate results. It is fully automatic and relies on gradient descent to optimize the calibration parameters. In the future, we expect to address some limitations of the method by calculating the poses of the reference sensor automatically instead of relying on the given ground truth, and by finding a way to bypass the need of priors for the other sensors. We would also like to add larger compatibility to other types of sensor, such as rolling shutter cameras or distorted LiDARs, and the optimization of intrinsic parameters. Finally, we would implement the multi-scene optimization, which should improve robustness by relying on more varied scenes to optimize a specific multi-sensor system, as well as the ability to manage dynamic elements in the scene, which are not considered currently.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[2] Jaewoong Choi, Junyoung Lee, Dongwook Kim, Giacomo Soprani, Pietro Cerri, Alberto Broggi, and Kyongsu Yi. Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2012. 1

[3] Andreas Geiger, Frank Moosmann, Ömer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *IEEE international conference on robotics and automation (RA-L)*, 2012. 1, 2

[4] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dex-NeRF: Using a neural radiance field to grasp transparent objects. In *Conference on Robot Learning (CoRL)*, 2020. 3

[5] Ganesh Iyer, R Karnik Ram, J Krishna Murthy, and K Madhava Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 2

[6] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[7] Jonathan Kelly and Gaurav S Sukhatme. A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors. In *Experimental Robotics: The 12th International Symposium on Experimental Robotics*, 2014. 2

[8] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2022. 5

[9] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 5

[10] Xudong Lv, Boya Wang, Ziwen Dou, Dong Ye, and Shuo Wang. Lccnet: Lidar and camera self-calibration using cost volume network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[11] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 2, 3

[12] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning (CoRL)*, 2022. 3

[13] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 2022. 3, 5

[14] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics (T-RO)*, 2015. 3

[15] Ashley Napier, Peter Corke, and Paul Newman. Cross-calibration of push-broom 2d lidars and cameras in natural scenes. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013. 1, 2

[16] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[17] Chanoh Park, Peyman Moghadam, Soohwan Kim, Sridha Sridharan, and Clinton Fookes. Spatiotemporal camera-lidar calibration: A targetless and structureless approach. *IEEE Robotics and Automation Letters (RA-L)*, 2020. 2, 3

[18] Joern Rehder, Paul Beardsley, Roland Siegwart, and Paul Furgale. Spatio-temporal laser to visual/inertial calibration with applications to hand-held, large scale scanning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014. 1, 2

[19] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[20] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. In *IEEE intelligent vehicles symposium (IV)*, 2017. 2

[21] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[22] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985. 3

[23] Hannes Sommer, Raghav Khanna, Igor Gilitschenski, Zachary Taylor, Roland Siegwart, and Juan Nieto. A low-cost system for high-rate, high-accuracy temporal calibration for lidars and cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 2

[24] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew Davison. iMAP: Implicit mapping and positioning in real-time. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[25] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 2020. 3

[26] Zachary Taylor and Juan Nieto. A mutual information approach to automatic calibration of camera and lidar in natural environments. In *Australian Conference on Robotics and Automation*, 2012. 2

[27] Zachary Taylor and Juan Nieto. Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Transactions on Robotics (T-RO)*, 2016. 2

[28] Rafael Peixoto Derenzi Vivacqua, Massimo Bertozzi, Pietro Cerri, Felipe Nascimento Martins, and Raquel Frizera Vassallo. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2017. 1

[29] Michael Waechter, Nils Moehrle, and Michael Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *European Conference on Computer Vision (ECCV)*, 2014. 3

[30] Fusang Wang, Arnaud Louys, Nathan Piasco, Moussab Bennehar, Luis Roldão, and Dzmitry Tsishkou. Planerf: Svd unsupervised 3d plane regularization for nerf large-scale scene reconstruction. *arXiv preprint arXiv:2305.16914*, 2023. 5

[31] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 2004. 5

[32] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance

fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 2, 3, 4

[33] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[34] Chongjian Yuan, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments. *IEEE Robotics and Automation Letters (RA-L)*, 2021. 1, 2, 6

[35] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004. 1, 2