2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Targetless Rotational Auto-Calibration of Radar and Camera for Intelligent Transportation Systems

Christoph Schöller*[12], Maximilian Schnettler*[12], Annkathrin Krämmer[12], Gereon Hinz[12], Maida Bakovic[12], Müge Güzet[12] and Alois Knoll[2]

***Abstract*— Most intelligent transportation systems use a combination of radar sensors and cameras for robust vehicle perception. The calibration of these heterogeneous sensor types in an automatic fashion during system operation is challenging due to differing physical measurement principles and the high sparsity of traffic radars. We propose – to the best of our knowledge – the first data-driven method for automatic rotational radar-camera calibration without dedicated calibration targets. Our approach is based on a coarse and a fine convolutional neural network. We employ a boosting-inspired training algorithm, where we train the fine network on the residual error of the coarse network. Due to the unavailability of public datasets combining radar and camera measurements, we recorded our own real-world data. We demonstrate that our method is able to reach precise and robust sensor registration and show its generalization capabilities to different sensor alignments and perspectives.**

I. Introduction

Modern intelligent transportation systems (ITS) utilize many redundant sensors to obtain a robust estimate of their perceived environment. By using sensors of different modalities, the system can compensate the weaknesses of one sensor type with the strengths of another. Especially in the field of traffic surveillance and ITS, the combination of cameras and radar sensors is common practice [1], [2], [3]. Reliably fusing measurements from such sensors requires precise spatial registration and is necessary to construct a consistent environment model. A precise sensor registration can be achieved with an extrinsic calibration that results in the correct transformation between the reference frames of the sensors in relation to each other and the world. Fig. 1 demonstrates the effects of an accurate sensor calibration. The upper image shows uncalibrated sensors, where the projected radar detections do not align with the vehicles. After the extrinsic calibration each detection overlays with its corresponding object in the image.

Manual sensor calibration is a tedious and expensive process. Especially in multi-sensor systems automatic calibration is crucial to handle the growing number of redundant sensors. Here manual calibration does not scale. Additionally, this technique is infeasible for automatic online recalibration, which is necessary to account for changes to the sensor system. These decalibrations occur frequently in real world applications, for example due to vibrations, wear and tear of the sensor mounting, or changing weather conditions. Furthermore, these calibration methods should be independent of explicitly provided calibration targets, as their installation into such systems or their observed scenes is impractical and would suffer from deterioration as well.

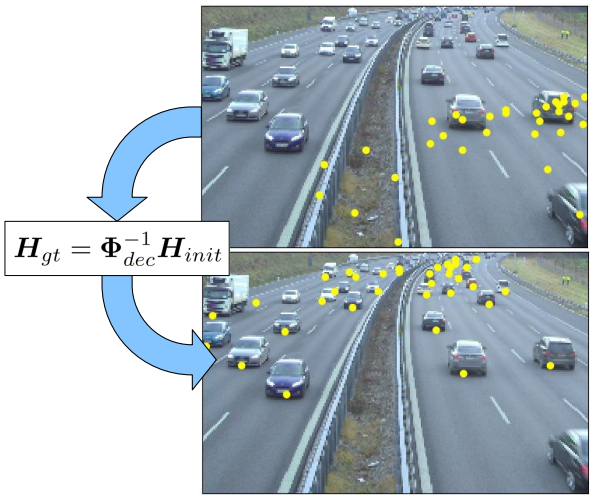


Fig. 1: The problem we solve is to estimate the correction $\boldsymbol{\Phi}_{dec}^{-1}$ of the initially erroneous calibration $\boldsymbol{H}_{init}$ that leads to the true transformation $\boldsymbol{H}_{gt}$ between the radar and the camera frame. This aligns the radar's vehicle detections (yellow points) with the vehicles in the camera image.

Calibrating systems with radar sensors and cameras is challenging due to different physical measurement principles and the high sparsity of radar detections. For the calibration without specific targets, a complex association problem between the sensors' measurements must be solved. As traffic radars do not provide visual features, such as edges, corners or color to easily associate detections with vehicles in the camera image, this association problem must be solved solely based on the relative spatial alignment and estimated distance measures between the vehicles.

In this paper we present – to the best of our knowledge – the first method for the automatic calibration of radar and camera sensors without explicit calibration targets. We focus on the rotational calibration between sensors because of its high influence on the spatial registration between cameras and radar sensors in ITS, especially for large observation distances. On the other hand, the projective error caused by translational miscalibrations in the centimeter range is negligible and easy to minimize in static scenarios by

This work has been funded by the German Federal Ministry of Transport and Digital Infrastructure as part of the project Providentia.

* These authors contributed equally to this work
[1]fortiss GmbH, Munich, Germany
[2]Technical University of Munich, Munich, Germany

measuring with modern laser distance meters. To solve the problem of rotational auto-calibration, we propose a two-stream Convolutional Neural Network (CNN) that is trainable in an end-to-end fashion. We employ a boosting-inspired training algorithm, where we first train a coarse model and afterwards transform the training data with its correction estimates to train a fine model on the residual rotational errors. We evaluate our approach on real-world data, recorded on the German highway A9 and show that it is able to achieve precise sensor calibration. Furthermore, we demonstrate the generalization capability of our approach by applying it to a previously unobserved perspective.

II. Related Work

Much research has been done on calibrating multi-sensor systems with homogeneous sensors (e.g., camera to camera), resulting in various state-of-the-art target-based and targetless calibration methods. However, it is a challenging problem to calibrate heterogeneous sensors with different physical measurement principles. While camera images provide dense data in form of pixels, lidar and even more so traffic radar sensors only record sparse depth data without color information. In this case it is difficult to match corresponding features between the sensors' measurements for calibration.

Classic approaches for the calibration of camera and laser-based depth sensors use planar checkerboards as dedicated calibration targets [4], [5]. These techniques achieve very precise estimates of the relative sensor poses, but require prepared calibration scenes. Approaches without physical targets calibrate the sensors by matching features of the natural scenes. In manual methods, a human has to pair the corresponding features in the image and depth data by hand [6]. For automatic classic approaches the capabilities regarding decalibration ranges and parameter extraction are limited [7], [8]. These drawbacks restrict their application scope and prevent them from being used for automatic calibration during system operation, which is possible with our method.

Recently, the task of sensor calibration has been approached using deep learning techniques. Early applications of CNNs in this field focus on camera relocalization [9]. With RegNet, Schneider et al. [10] presented the first CNN for camera and lidar registration, which performs feature extraction, matching, and pose regression in an end-to-end fashion on an automotive sensor setup. Their method is able to estimate the extrinsic parameters and to compensate decalibrations online during operation. To refine their result the authors use multiple networks, trained on datasets with different calibration margins. In contrast, we do not need to define calibration margins for our networks, as our second network specializes on the errors of the first network by design. Liu et al. [11] apply this method to the calibration of three sensors by first fusing a depth camera and a lidar that were calibrated with RegNet, and then they use the resulting dense point cloud for the calibration to a camera. Iyer et al. [12] propose CalibNet, which they train with a geometric and photometric consistency loss of the input images and point clouds, rather than the explicit calibration parameters. Due to difficulties in estimating translation parameters in a single run, they first estimate the rotation and use it to correct the depth map alignment. Then they feed the corrected depth map back into the network to predict the correct translation. However, in contrast to our approach these methods use lidar sensors with relatively dense point clouds compared to the measurements of traffic radars.

The measurement characteristics of radars cause a lack of targetless calibration methods. Traffic radars output preprocessed measurement data in form of detected objects. They lack descriptive visual features and are sparse. Additionally, measurement noise, missing object detections, and false positives make the calibration of radars with sensors of different modality particularly challenging. Existing approaches for the calibration of multi-sensor systems with radars rely on dedicated targets, such as corner reflectors or plates, based on conductive material that ensures reliable radar detections [13], [14]. Recently, these calibration concepts were extended towards the combination of radars with other sensor types. Especially the calibration with cameras is challenging, as the sensors do not share common features such as color, shapes or depth. Natour et al. [15] calibrate a radar-camera setup by optimizing a non-linear criterion, obtained from a single measurement with multiple targets and known inter-target distances. However, the targets in the radar and image data are extracted and matched manually. Peršić et al. [16] designed a triangular target to calibrate a 3D lidar and an automotive radar. They experienced variable error margins in the estimated calibrations due to the sparse and noisy radar data and the geometric properties of their sensor setup. As a result, an additional optimization step using a priori knowledge of the specified radar field-of-view refines these estimated parameters. Song et al. [17] use a radar-detectable augmented reality marker for a traffic surveillance system based on a 2D radar and camera, enabling an analytic solution of the paired measurements. However, there is a lack of approaches for automatic and targetless radar-camera calibration which we address in this work.

III. Problem Statement

To calibrate a radar and camera to each other, the transformation that correctly projects the radar detections into the camera image must be estimated. This is the case when each projected detection spatially aligns with its corresponding object in the image. As we use a traffic radar, the detected objects are vehicles as shown in Fig. 1. However, our approach is not limited to the traffic domain.

The described projection of detections into the image can be computed by

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \boldsymbol{K}\boldsymbol{H}\boldsymbol{x}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^3$ is the position of a detected vehicle in the radar coordinate system, $[u, v]^T$ are its corresponding pixel coordinates and z_c the straight-line distance of the detected

vehicle to the image plane of the camera, i.e. the depth of the projected pixel. The projection matrix $\boldsymbol{K} \in \mathbb{R}^{3\times4}$ is based on the intrinsic camera parameters and $\boldsymbol{H} \in \mathbb{R}^{4\times4}$ is the extrinsic calibration matrix. The latter represents the camera pose relative to the radar and is defined as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \tag{2}$$

with $\boldsymbol{R} \in SO(3)$ being the rotational and $\boldsymbol{t} \in \mathbb{R}^3$ the translational component. While $\boldsymbol{K}$ can be estimated in a controlled calibration setting prior to deploying the sensors, $\boldsymbol{H}$ must be determined after deployment in situ. In our work we focus on computing the rotational component as – compared to the translational component – it is hard to measure and has a high impact on the quality of the inter-sensor registration, especially for large observation distances.

Our goal is to estimate the transformation $\boldsymbol{H}_{gt}$, that describes the true relative pose between the two sensors. A correct estimate results in the alignment of projected radar measurements and vehicles in the image. Assuming an initially incorrect calibration $\boldsymbol{H}_{init}$, we need to determine the present decalibration transformation $\boldsymbol{\Phi}_{dec}$ that represents the error between $\boldsymbol{H}_{init}$ and $\boldsymbol{H}_{gt}$ and thus

$$\boldsymbol{H}_{init} = \boldsymbol{\Phi}_{dec}\boldsymbol{H}_{gt}. \tag{3}$$

In fact, we directly estimate $\boldsymbol{\Phi}_{dec}^{-1}$, since it can be used to recover the correct calibration $\boldsymbol{H}_{gt}$ without additionally inverting $\boldsymbol{\Phi}_{dec}$.

IV. Our Approach

Our objective is to regress the relative orientation of a camera with respect to a radar sensor. To achieve this, an association problem between the radar detections and the vehicles in the camera image must be solved. This is a difficult problem, as radar detections do not contain descriptive features. A neural network can learn how to solve this association problem based on the spatial alignment between the projected radar detections and the vehicles in the image.

Our approach leverages two convolutional neural networks, where we train the first coarse network on the initially decalibrated data and then a fine network on its residual error. Both models share the same architecture, loss and hyperparameters. In this section we first explain the model and then the training process in detail.

A. Architecture

Our model is built as a two-stream neural network, consisting of a rgb-input and a radar-input as shown in Fig. 2. It outputs a transformation to correct the rotational error of the calibration between respective camera and radar as a quaternion.

The rgb-input is a camera image and the radar-input is a sparse matrix with radar projections. The image is standardized and resized to a resolution of 240×150 pixels. It gets propagated through the rgb stream of our network that starts with a cropped MobileNet [18] with width multiplier 1.0. We crop the MobileNet after the third depthwise convolution layer (conv_dw_3_relu) to extract low-level features, while preserving spatial information. We use a MobileNet that has been pre-trained on ImageNet [19], but include the layers for fine-tuning in further training. The MobileNet is followed by two MlpConv [20] layers, each consisting of a 2D convolution with kernel size 5×5, followed by two 1×1 convolutions and 16 filter maps in each component. The task of the rgb stream is to detect vehicles and to estimate where radar detections will occur.

The radar-stream receives the projected radar detections with the same resolution as the camera image as input. Each projection occupies one cell in the sparse matrix and stores the inverse depth $1/z_c$ of respective projected detection, as proposed by [10]. We apply a 2×2 max-pooling to reduce the input dimension to a feasible size and do not use convolutions in the radar stream to retain the sparse information.

Then we embed each stream into a 50 dimensional latent vector using a fully-connected layer. This latent vector contains the input information in a dense and compressed format. The following regression block consists of three layers with 512, 256 and 4 neurons. Between the first two layers we apply dropout regularization [21]. The four output neurons correspond to the components of the quaternion that describes the calibration correction. We use linear activations for the final output layer, and PReLu [22] activations everywhere else, except in the MobileNet block. This empirically lead to better performance compared to classic ReLu activations. The task of the regression block is to estimate the rotational correction that solves the misalignment between the camera image and the radar detections.

B. Loss Function

We use the Euclidean distance as the loss function between the true quaternion $\boldsymbol{q}$, and the predicted quaternion $\hat{\boldsymbol{q}}$ that represents the estimated correction of the decalibration.

The Euclidean distance is a common distance measure to define a rotational loss function over quaternions [10], [9]. Since this metric is ambiguous and can lead to different errors for the same rotations [23], we also evaluated the performance of our approach using the geodesic quaternion metric

$$\mathcal{L}_\theta = 1 - |\boldsymbol{q} \cdot \frac{\hat{\boldsymbol{q}}}{\|\hat{\boldsymbol{q}}\|}| + \alpha|1 - \|\hat{\boldsymbol{q}}\|\,|, \tag{4}$$

proposed by [24]. We added a length error term, weighted by α that we empirically evaluated to 0.005. Without this additional length term the network's output diverges and the learning plateaus. As this loss resulted in similar performance despite its theoretical superiority, we finally used the Euclidean distance $\|\boldsymbol{q} - \hat{\boldsymbol{q}}\|$ to save an additional hyperparameter.

C. Hyperparameters

We use the Adam optimizer [25] with the parameters proposed by its authors and learning rate 0.002, that we reduce by a factor of 0.2 once the validation loss plateaus for five epochs. To initialize our weights we use orthogonal

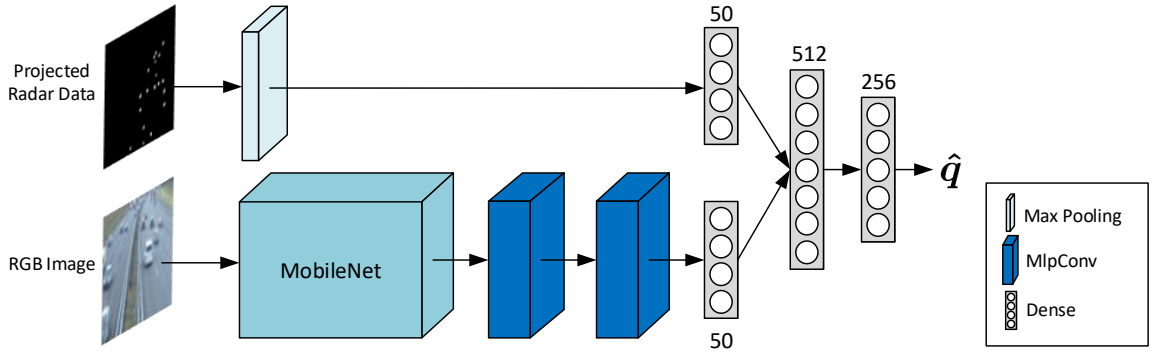


Fig. 2: The architecture of our network consists of two input streams, one for radar projections and one for rgb images. Both streams end in a 50 neuron fully-connected layer to condense information. Then we regress for an output quaternion $\hat{\boldsymbol{q}}$ that describes the calibration correction.

initialization [26]. For the dropout we set a probability of 0.5, use a batch size of 16 and early stopping when the validation loss does not improve for 10 epochs.

D. Cascaded Residual Learning

To improve the calibration results of our first, coarse network that we train on the original data, we train a second, fine network on the remaining residual error. This is inspired by gradient boosting algorithms [27], where each subsequent learner is trained on the residual error of the previous one.

This boosting has multiple advantages that lead to more accurate calibration parameters. During operation, the first network roughly corrects the initial calibration error and the sensors are approximately aligned. For the second network more radar detections can be projected into the camera image, which leads to a higher number of correspondences that enable the second network to perform a more fine-grained correction. Furthermore, the second network implicitly focuses on solving the errors in those axes that the first network performed poorly on. In our case, the fine network performs much better on solving the roll error, as errors around the z-axis of the camera cause only relatively small projective discrepancy, and thus the coarse network focuses on tilt and pan.

In detail, we train the first network on dataset D, for which the radar detections of each sample were projected with a transformation $\boldsymbol{H}_{init,i}$, obtained by applying a random decalibration $\boldsymbol{\Phi}_{dec,i}$ to the true calibration $\boldsymbol{H}_{gt}$. Index i refers to the sampled decalibration. After the first network's training we transform D into a new dataset D', on which we train the second network. D' contains training samples corrected by the output of the first network. We transform the samples by converting the output quaternion $\hat{\boldsymbol{q}}_i$ for each sample to transformation $\hat{\boldsymbol{\Phi}}^{-1}_{dec,i}$. Then we compute a new, corrected extrinsic matrix

$$\hat{\boldsymbol{H}}_{gt,i} = \hat{\boldsymbol{\Phi}}^{-1}_{dec,i}\boldsymbol{H}_{init,i} \tag{5}$$

for each sample and reproject the corresponding radar detections as described by Eq. 1. We obtain the correction of the residual decalibration by

$$\boldsymbol{\Phi}'^{-1}_{dec,i} = \boldsymbol{\Phi}^{-1}_{dec,i}\hat{\boldsymbol{\Phi}}_{dec,i}, \tag{6}$$

which serves as the new label in the transformed dataset D'. The second network is then trained on D'.

At inference time we obtain the approximate true calibration for a new sample by computing

$$\hat{\boldsymbol{H}}_{gt} = \hat{\boldsymbol{\Phi}}'^{-1}_{dec}\hat{\boldsymbol{\Phi}}^{-1}_{dec}\boldsymbol{H}_{init}, \tag{7}$$

where $\hat{\boldsymbol{\Phi}}'^{-1}_{dec}$ is the output of the second network and $\hat{\boldsymbol{\Phi}}^{-1}_{dec}$ of the first network. Note that before computing $\hat{\boldsymbol{\Phi}}'^{-1}_{dec}$ we perform a reprojection in the same way as during training.

E. Iterative and Temporal Refinement

In the field of ITS and autonomous driving, sensor data is usually available as a continuous stream. A single decalibration of the sensor setup is more likely than completely random decalibrations for each sample. A temporal average over correction estimates for multiple consecutive samples can reduce the influence of estimation errors made for individual samples and thus increase the robustness and accuracy of our method.

V. Experiments

In this section we explain which data we used to train and evaluate our approach. Furthermore, we explain our evaluation process in detail and present quantitative, as well as qualitative results.

A. Dataset

In the field of ITS, public datasets containing data of radars combined with cameras are not available. Therefore, we generated our own dataset using sensor setups developed within the scope of the research project Providentia [3]. Two identical setups were installed on existing gantry bridges along the Autobahn A9, overlooking a total of eight traffic lanes. Our sensor setup is shown in Fig. 3 and consists of a Basler acA1920-50gc camera with a lens of $25\,\mathrm{mm}$ focal length and a smartmicro UMRR-0C Type 40 traffic radar.

The camera records rgb images with a resolution of 1920×1200 pixels, while the radar outputs vehicle detections as positions. The radar measurements can result in undetected vehicles, multi-detections for large vehicles like trucks or buses, and false positives due to measurement noise.

Fig. 3: Our sensor setup with a radar and camera above the highway.

1) Ground Truth Calibration: Since our approach requires a reference transformation $\boldsymbol{H}_{gt}$ between the sensors, we put special effort and care on the initial manual calibration. This is equivalent with manual labeling in other supervised learning problems.

We calibrated the cameras intrinsically with a checkerboard based method in our laboratory, while the radar is intrinsically calibrated ex-factory. The translational extrinsic parameters of the sensor setup were manually measured on-site with a spirit level and laser distance meter. We estimated the initial rotation parameters of the sensors with respect to the road using vanishing point based camera calibration [28] (one vanishing point, known height above the road and known focal length) and the internal calibration function of the radar sensor. Afterwards, we fine-tuned the extrinsic rotational parameters by minimizing the visual projective error.

2) Training Data Generation: To obtain the necessary number of samples to train and evaluate our networks, it would be infeasible to record with many different sensor setups and manually determine each groundtruth calibration. Therefore, we randomly distorted the calibration $\boldsymbol{H}_{gt}$ for one sensor setup per measurement point as proposed by [10]. In particular, we randomly generated 6-DoF decalibrations $\boldsymbol{\Phi}_{dec}$ for each sample and used these decalibrations to compute initial decalibrated extrinsic matrices $\boldsymbol{H}_{init}$, according to Eq. 3. Afterwards, we projected the radar detections on the image according to Eq. 1, leading to a mismatch between the detections and the vehicles in the image. Besides, we filtered generated samples with less than 10 remaining correspondences. This ensures the exclusion of training samples without correspondences between camera and radar projection, with which learning is not possible.

In particular, the decalibration angles were sampled from a uniform distribution on $[-10°, 10°]$ for the tilt and pan, and $[-5°, 5°]$ for the roll angle. We assumed a smaller roll decalibration as this angle is easier to measure with a spirit level. We multiplied resulting matrices into a single rotational decalibration. Furthermore, we added a translation error with a standard deviation of $10\,\mathrm{cm}$. Even though translation errors are minimal as distances are easy to measure, by this we account for errors during the manual calibration process and show that our approach is robust to it. Creating our dataset as described resulted in a total of 37929 samples for the first sensor setup, of which we used 34137 for training and 3792 for validation. Additionally, we generated an independent test set $\mathcal{T}_1$ with 2536 samples, where we only included images and radar detections that do not appear in the training data. We further generated a second test set $\mathcal{T}_2$ with 2012 samples from a different sensor setup on a second gantry bridge in the same manner to evaluate the generalization of our approach.

B. Evaluation

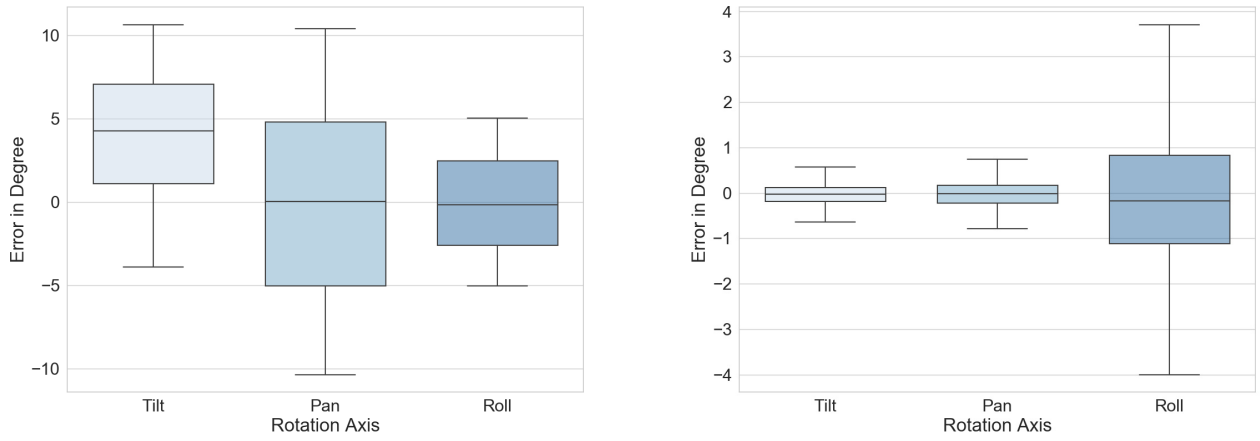
We trained our models with the boosting-inspired approach described in Sec. IV-D and the dataset generated with random decalibrations as explained in Sec. V-A.

1) Random Decalibration: Tab. I shows the average angular errors of our networks using test set $\mathcal{T}_1$ with random decalibrations. While the coarse network achieves significant improvements in the tilt and pan angles, it struggles to correct the roll. The roll calibration error is weaker correlated with the input as it has only a small projective influence over the long distances we work with. However, our fine model decreases the roll error significantly as it has more influence on the total remaining projective discrepancy after the coarse correction step. In total, we achieve a mean error reduction of $95.3\,\%$ in tilt, $93.5\,\%$ in pan and $47.7\,\%$ in roll over the initial decalibrations. The remaining errors after our correction are approximately normally distributed around zero, which means our approach works reliably with only few outliers and can be trusted in a real-world setting (Fig. 4).

In Fig. 5 we demonstrate qualitative examples of applying our approach to different decalibration scenarios. The main task of the coarse network is to find the right correspondences between radar detections and vehicles in the images. Based on these correspondences it estimates a rough correction for the initial decalibration. In case of decalibrations with only few successfully projected detections, the network's correction leads to more projections onto the image plane that are then provided to the fine network. This effect can be observed in the first row in Fig. 5. The fine network makes use of the increased number of correspondences and refines the calibration as shown in column **(c)**. It is particularly good at correcting rotational errors in the roll direction. In the second row **(b)** it can be observed that the coarse network is not able to solve the roll error because it has a relatively small impact on the projection discrepancy. The yellow points in

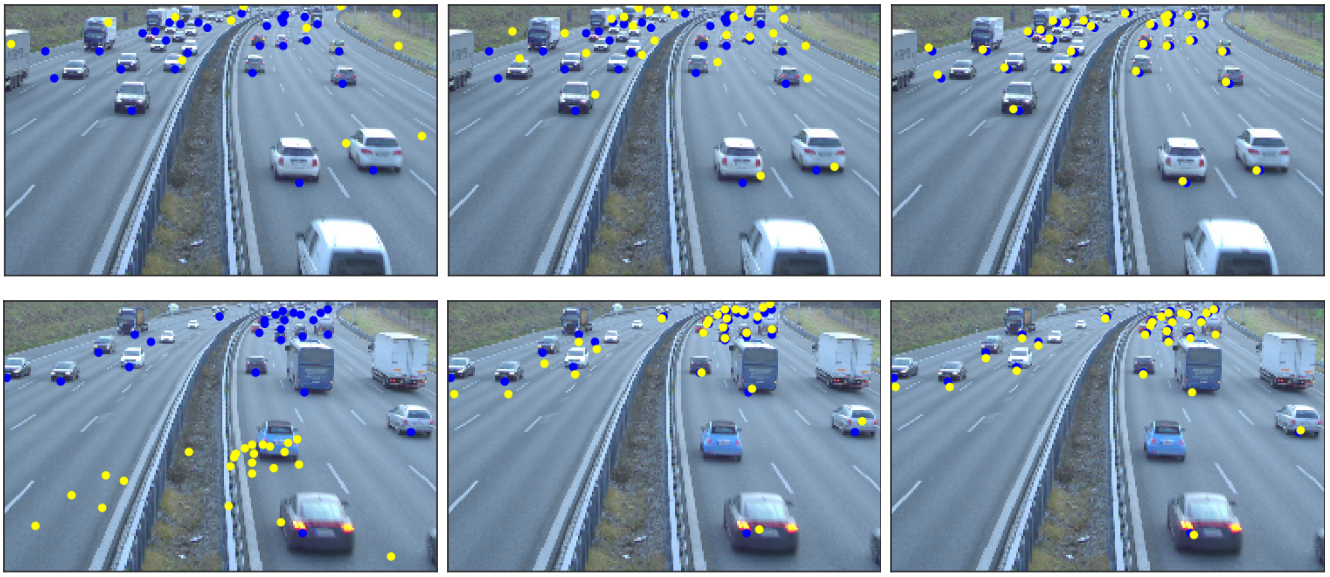
	Tilt	Pan	Roll	Total
Initial	$4.45°$	$4.95°$	$2.52°$	$7.90°$
Coarse Network	$0.46°$	$0.81°$	$2.43°$	$2.78°$
Fine Network	**0.21°**	**0.32°**	**1.32°**	**1.45°**

TABLE I: Mean absolute errors for all axes and in total initially, after applying the coarse network and after applying the fine network on the test set $\mathcal{T}_1$. The fine model focuses on correcting the remaining roll error and significantly improves tilt and pan as well.



(a) Initial **(b)** Calibration Result

Fig. 4: Initial and resulting errors for each rotation angle on test set $\mathcal{T}_1$ with random calibration errors. Note that the positive shift for the tilt angle errors is a result of filtering the samples with less than 10 projected detections in the image, as described in Sec. V-A. Tilting the camera downwards likely moves the detections out of the upper image border.

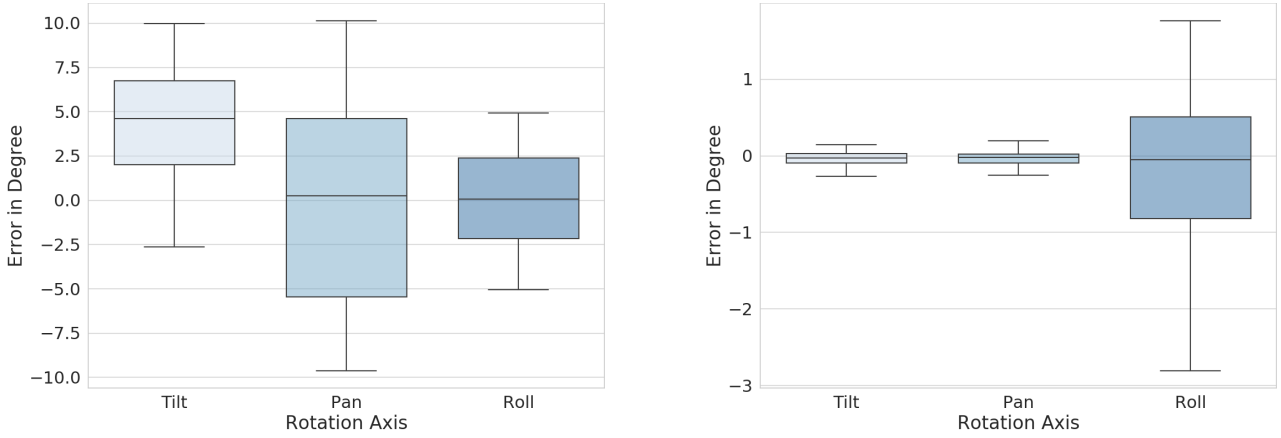


(a) Initial **(b)** Coarse Network **(c)** Fine Network

Fig. 5: Each row depicts the application of our model to a decalibrated sample from test set $\mathcal{T}_1$. Blue points represent the projected radar detections using the ground truth calibration and yellow projections using the calibration at (a) the initial stage, (b) after applying the coarse network and (c) after applying the fine network. While the coarse network achieves a reasonable, but still imprecise calibration, the fine model handles the precise adjustment.

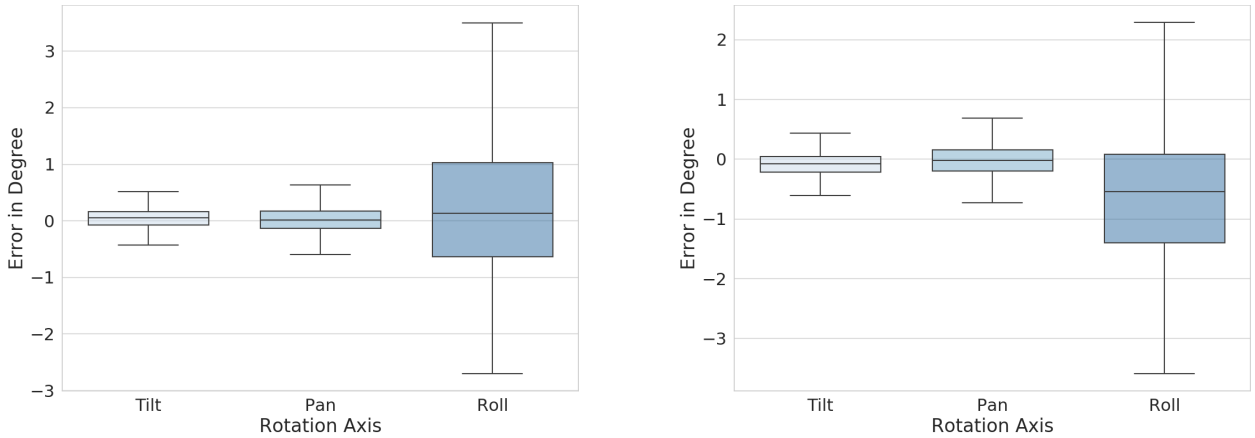
the left half of the image are rotated below, and in the right half of the image above the blue ground truth detections. The fine network in the second row **(c)** was able to correct this residual error.

2) Static Decalibration: We also evaluated our approach by applying the same decalibration to all samples of the test set $\mathcal{T}_1$. This is a more realistic setting. In this manner we evaluated 100 different decalibrations. In particular, we computed the error for each decalibration as the mean error over all samples. This way our approach achieved on average decalibration errors of $0.21°$ for tilt, $0.35°$ for pan and $1.33°$ for roll. As shown in Fig. 6 **(b)**, taking the average over all sample errors with the same static decalibration significantly reduces the error variance compared to using only a single frame for calibration like in the random decalibration setting shown in Fig. 4 **(b)**. Our model is able to reduce the static errors over all samples towards a distribution with approximately zero mean, as shown for two examples in Fig. 7. This indicates that temporal averaging of the estimated decalibration corrections as proposed in Sec. IV-E could be a suitable method to further improve accuracy and robustness.



(a) Initial **(b)** Result

Fig. 6: Initial and resulting distributions of the errors for the 100 static decalibrations for test set $\mathcal{T}_1$. For each decalibration we averaged all sample errors.



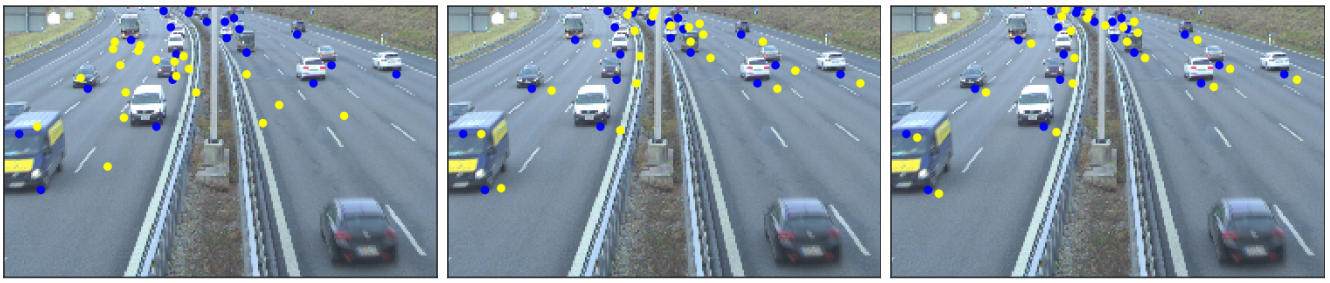
(a) Initial errors: tilt = $3.21°$, pan = $6.24°$, roll = $1.75°$ **(b)** Initial errors: tilt = $7.46°$, pan = $-7.99°$, roll = $-2.31°$

Fig. 7: Resulting error distributions over all $\mathcal{T}_1$ test set samples for two different static decalibrations. The means are close to zero, which shows that a temporal averaging could further improve the performance of our approach.

3) Generalization: To demonstrate the generalization capability of our approach we applied it to test set $\mathcal{T}_2$, which is obtained from a sensor setup located at a different gantry bridge that was not included in the training data and has never been observed before. In this case the trajectory of the street is different and thus the distribution of vehicles in the image. Besides, the true extrinsic calibration differs from the first sensor setup and the perspective of the camera observing the vehicles changed. Despite these challenges, our approach achieved reasonable results for random decalibrations with average errors of $0.36°$ for tilt, $1.88°$ for pan and $2.83°$ for roll (Fig. 8). While the performance dropped compared to the sensor setup used for training, it indicates that our approach is able to generalize if trained on a more diverse dataset with different perspectives and road segments. Furthermore, the achieved results already vastly reduce manual calibration efforts. It can also support other calibration methods in practice, as the difficulty to match correspondences is greatly reduced.

VI. Conclusion

The manual calibration of sensors in an ITS is tedious and expensive, especially concerning sensor orientations. For radars and cameras there is a lack of automatic calibration methods due to the sparsity and absence of descriptive features in radar detections. We addressed this problem and presented the first approach for automatic rotational calibration of radar and camera sensors without the need of dedicated calibration targets. Our approach consists of two convolutional neural networks that are trained with a boosting-inspired learning regime. We evaluated our method on a real-world dataset that we recorded on a German highway. Our method achieves precise rotational calibration of the sensors and is robust to missing vehicle detections, multiple detections for single vehicles and noise. We demonstrated its generalization capability and achieved reasonable results by applying it on a second measurement point with a different viewing angle on the highway and vehicles. This drastically reduces the efforts of manual calibration.



(a) Initial **(b)** Coarse Network **(c)** Fine Network

Fig. 8: Calibration results of applying our approach which was trained on one sensor setup to an unseen sensor setup (test set $\mathcal{T}_2$). Blue points represent the projected radar detections using the ground truth calibration and yellow projections using the calibration at (a) the initial stage, (b) after applying the coarse network and (c) after applying the fine network. Even though our model has never observed this perspective during training and the true calibration differs from the one in the training data, it generalizes and achieves reasonable results.

We expect that in the future the generalization capabilities of our approach could be further improved by using a more diverse dataset that includes multiple camera perspectives. Furthermore, as sensors record a time series, sequences of frames could be used for iterative calibration with a recurrent neural network to increase calibration precision and robustness. As after the application of our approach the association of radar detections with vehicle detections in the image can be easily achieved with nearest-neighbor algorithms, the final results could be revised by solving a classic, convex optimization problem.

REFERENCES

[1] M. Wang, L. Jiang, W. Lu, and Q. Ma, "Detection and tracking of vehicles based on video and 2D radar information," *International Conference on Intelligent Transportation (ICIT)*, 2017.

[2] A. Roy, N. Gale, and L. Hong, "Fusion of doppler radar and video information for automated traffic surveillance," *International Conference on Information Fusion (FUSION)*, 2009.

[3] G. Hinz, M. Buechel, F. Diehl, G. Chen, A. Kraemmer, J. Kuhn, V. Lakshminarasimhan, M. Schellmann, U. Baumgarten, and A. Knoll, "Designing a far-reaching view for highway traffic scenarios with 5G-based intelligent infrastructure," *8. Tagung Fahrerassistenz*, 2017.

[4] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," *International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[5] G. Zhi, Z. Sidong, Z. Wei, and Z. Yunyi, "A high-precision calibration technique for laser measurement instrument and stereo vision sensors," *International Conference on Electronic Measurement and Instruments (ICEMI)*, 2007.

[6] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes," *International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[7] H.-J. Chien, R. Klette, N. Schneider, and U. Franke, "Visual odometry driven online calibration for monocular lidar-camera systems," *International Conference on Pattern Recognition (ICPR)*, 2016.

[8] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers," *Robotics: Science and Systems (RSS)*, 2013.

[9] A. Kendall, M. K. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," *International Conference on Computer Vision (ICCV)*, 2015.

[10] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," *Intelligent Vehicles Symposium (IV)*, 2017.

[11] H. Liu, Y. Liu, X. Gu, Y. Wu, F. Qu, and L. Huang, "A deep-learning based multi-modality sensor calibration method for usv," *International Conference on Multimedia Big Data (BigMM)*, 2018.

[12] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Self-supervised extrinsic calibration using 3D spatial transformer networks," *International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[13] R. E. Helmick and T. R. Rice, "Removal of alignment errors in an integrated system of two 3-D sensors," *Transactions on Aerospace and Electronic Systems (T-AES)*, 1993.

[14] Z. Li and H. Leung, "An expectation maximization based simultaneous registration and fusion algorithm for radar networks," *Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2006.

[15] G. E. Natour, O. A. Aider, R. Rouveure, F. Berry, and P. Faure, "Radar and vision sensors calibration for outdoor 3D reconstruction," *International Conference on Robotics and Automation (ICRA)*, 2015.

[16] J. Peršić, I. Marković, and I. Petrović, "Extrinsic 6DoF calibration of 3D lidar and radar," *European Conference on Mobile Robots (ECMR)*, 2017.

[17] C. Song, G. Son, H. Kim, D. Gu, J. H. Lee, and Y. Kim, "A novel method of spatial calibration for camera and 2D radar based on registration," *International Congress on Advanced Applied Informatics (AAI)*, 2017.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861 [cs.CV]*, 2017.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[20] M. Lin, Q. Chen, and S. Yan, "Network in network," *International Conference on Learning Representations (ICLR)*, 2013.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research (JMLR)*, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," *International Conference on Computer Vision (ICCV)*, 2015.

[23] D. Q. Huynh, "Metrics for 3D rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, 2009.

[24] J. J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," *International Conference on Robotics and Automation (ICRA)*, 2004.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2014.

[26] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *International Conference on Learning Representations (ICRA)*, 2014.

[27] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics and Data Analysis (CSDA)*, 2002.

[28] N. K. Kanhere and S. T. Birchfield, "A taxonomy and analysis of camera calibration methods for traffic monitoring applications," *Transactions on Intelligent Transportation Systems (T-ITS)*, 2010.