CaRaCTO–3D: From Camera–Radar Calibration to Scene Reconstruction

Mahdi Chamseddine*[1,2], Jason Rambach[1], Didier Stricker[1,2]

[1]German Research Center for Artificial Intelligence (DFKI),
Kaiserslautern, Germany.
[2]RPTU Kaiserslautern-Landau, Kaiserslautern, Germany.

Contributing authors: firstname.lastname@dfki.de;

Abstract

The multimodal nature of camera and radar sensor data enables various automation and surveillance tasks, where one sensor compensates for the limitations of the other sensor: cameras capture high-resolution color data, while radar measures depth and velocity of targets. Calibration is essential to fuse these data modalities effectively. This work presents a robust extrinsic calibration algorithm for camera-radar setups, extending standard geometric constraints with elevation information to enhance optimization. Unlike existing methods, this approach relies solely on camera and radar data without requiring complex targets or external measurements. The 3D calibration enables the estimation of the target elevation which is lost when using 2D radar. We evaluate our results against a sub-millimeter ground truth system, demonstrating superior performance compared to more complex algorithms. Leveraging these accurate calibration results, we subsequently employ monocular depth estimation and instance segmentation techniques to perform camera-radar data fusion, allowing 3D target and scene reconstruction. github.com/mahdichamseddine/CaRaCTO.

Keywords: Calibration, Camera, Radar, Robotics, Segmentation, Reconstruction

∗ Corresponding author.
On behalf of all authors, the corresponding author states that there is no conflict of interest.

1 Introduction

Environment sensing plays a crucial role in various modern applications. Whether in robotics, surveillance [1, 2], autonomous driving, or assistive driving [3, 4], sensors such as cameras, radar, and lidar are employed to detect and classify objects and obstacles within the environment. These sensors possess distinct characteristics that complement rather than replace one another. Cameras offer high-resolution color imagery, texture, and contextual information, while lidar and radar provide depth and dimensional data. Although lidar data typically has a higher spatial density compared to radar data, radar is more resilient to adverse weather and lighting conditions and can measure velocities.

To achieve a comprehensive understanding of the environment, data from these different sensors are often fused. This fusion enables the detection of various objects and obstacles using multimodal features, such as dimensions, position, velocity, and orientation [5–8]. However, before sensor fusion can occur, a crucial calibration step is required to align data from all sensors within a common reference frame, ensuring accurate data association. This calibration is thus a fundamental step in any data processing problem.

Despite their advantages, lidar sensors remain expensive, limiting their commercial adoption compared to cameras and radar sensors, which have been in use for a longer time. Although high-resolution 3D radar sensors are gaining popularity [9, 10], 2D radar sensors are still the most widely used type of radar in commercial applications due to their lower cost. Therefore, the calibration method presented in this work focuses on a 2D radar and camera setup, taking into account their affordability and widespread use.

This paper introduces an extrinsic calibration algorithm for camera-radar systems. Unlike other approaches that project radar data onto a 2D plane, this method retains elevation information, using the camera to estimate and reconstruct targets in 3D. Additionally, the proposed approach aims to enhance the robustness of the optimization process against poor initialization, simplify the calibration setup, and make it more accessible, all while maintaining or improving upon the quality of existing algorithms.

Furthermore, we introduce a scene reconstruction pipeline that makes use of the extrinsic calibration as well as recent advancements in instance segmentation and monocular depth estimation to *fuse* the camera and radar data to generate a 3D point cloud of the scene. Scene reconstruction is a popular computer vision task where the 3D point cloud of a scene is estimated from a single or multiple images [11–13] and has several applications in augmented reality, robotics, autonomous driving, and construction. One challenging aspect of scene reconstruction is the absence of a metric scale reference, and this is where a depth sensing sensor such as the radar proves beneficial.

In the previous work [14] that this paper is building upon we presented the following contributions:

- Extrinsic camera-radar calibration that does not require external sensing.
- Improved optimization formulation for added robustness.

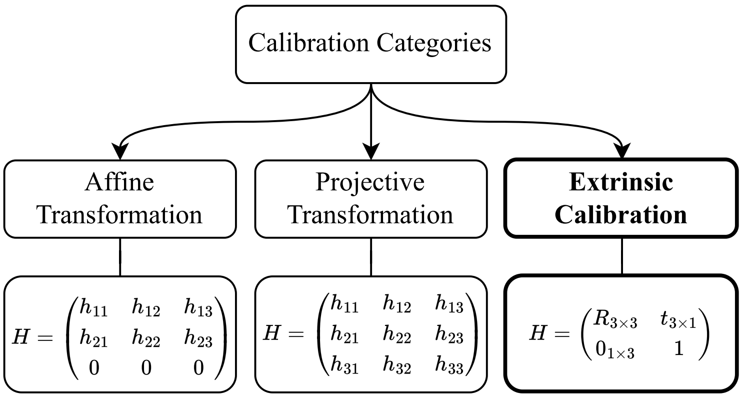

Calibration Categories
Affine Transformation
Projective Transformation
Extrinsic Calibration
$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 0 \end{pmatrix}$
$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$
$H = \begin{pmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix}$

Fig. 1: The different types of calibration as described by Oh et al. [15]. Our approach belongs to the third category (extrinsic calibration). (Figure from [14])

- Extensive evaluation showing stability and significant improvement.

We extend it to present the additional contributions:

- A method for camera-radar correspondence detection using an instance segmentation model.
- Full 3D scene reconstruction using a monocular depth estimation model combined with radar measurements for recovering metric scale.
- Extended experiments on robustness of the calibration algorithm and qualitative results of the target matching and 3D reconstruction.

The rest of the paper is structured as follows: Section 2 summarizes the related work and previous contributions to the field. In Section 3 we define the calibration problem and present our proposed approach. Section 4 presents the correspondences matching and our scene reconstruction pipeline. Sections 5 and 6 discuss the quantitative evaluation of the calibration and the qualitative results of the reconstruction. Finally, concluding remarks are given in Section 7.

2 Related Work

2.1 Camera-Radar Calibration

Multiple studies have been published on camera-radar calibration. A comparative work by Oh et al. [15] categorizes camera-radar calibration methods into three main types (see Figure 1): affine transformations, projective transformations, and extrinsic calibration, the latter of which is the focus of our work.

Wang et al. [6] and Kim et al. [8] propose methods where an affine transformation is computed between 2D radar points and their corresponding pixel locations in the image. In their approach, a pseudo-inverse is employed to solve a least-squares problem for the two-dimensional affine transformation. The quality of this transformation is evaluated by measuring the image distance between the transformed radar points and their corresponding image points.

Contrary to the 2D affine transformation calibration which estimates only six out of nine transformation parameters, the 2D projective transformation method estimates the complete 3×3 homography between the radar and camera planes. The use of projective transformation for camera-radar calibration was presented by Sugimoto et al. [5] and Wang et al. [7]. Sugimoto et al. filtered for the points that belong to the "radar measurement plane" to yield a more accurate calibration. Although those approaches can yield more accurate calibration results than affine transformations, they do not account for the 3D nature of the data, providing only point correspondences between the radar and camera planes.

The third category is extrinsic calibration, which can be further divided into two types: multi-sensor extrinsic calibration, involving camera, lidar, and radar, and camera-radar only extrinsic calibration.

Domhof et al. [16] treat radar data in 2D, using Euclidean error to solve the optimization and compute the extrinsic parameters. Additionally, they designed a complex joint target for camera, lidar, and radar calibration.

In contrast to other camera-radar calibration methods, Peršić et al. approach the problem with the assumption that 2D representations in both the image and radar data correspond to targets in 3D space. They first introduced a radar-lidar calibration [17] then extended it to also work with a radar-camera system [18]. To estimate the radar elevation they used the known radar cross section of the target.

Similarly, El Natour et al. [19] utilizes the distance between multiple targets to recover the full 3D representation from 2D sensors. This enables 3D reconstruction of targets once the system is calibrated. However, achieving this result requires multiple targets with accurately measured distances between them. The authors attempt to mitigate this limitation in [20] by moving the sensor system while keeping the targets fixed, and incorporating sensor trajectory estimation.

There are other methods that applied 3D extrinsic calibration using sensors capable of inherently capturing elevation information, making them less directly comparable to our setup. These include methods leveraging 3D radar or LiDAR, where elevation is measured rather than estimated. For instance, Wise et al. [10, 21] propose a continuous-time radar-to-camera calibration framework using 3D FMCW radar, which provides true elevation and Doppler measurements. Cheng et al. [22] introduce a flexible and accurate method for 3D radar-camera co-calibration using a target-based approach, which also assumes 3D radar data availability. Additionally, Wang et al. [23] present LVI-ExC, a target-free LiDAR-visual-inertial calibration framework that integrates elevation directly from LiDAR. While these methods are more recent and demonstrate high accuracy, they rely on sensing modalities that differ fundamentally from our 2D radar sensor, where elevation must be inferred during the calibration process itself.

Our work presents an extrinsic calibration algorithm designed to estimate the rotation and translation between camera and radar sensors using the 3D representation of the targets. Unlike previous approaches, our method requires only a single retroreflector and does not necessitate complex target designs and does not rely on the radar cross section value which is not always measured by the radar. Additionally, our algorithm is robust even with sub-optimal initialization, due to the incorporation of elevation constraints. The results are validated against previous works using a high-quality ground truth system, marking the first time such evaluations have been performed in this context.

2.2 Monocular Depth Estimation

Monocular depth estimation is a technique that enables depth perception from a single image, reducing hardware costs and complexity. It is essential for applications like autonomous driving [24], robotics [25], and augmented reality [26], where understanding the spatial layout of an environment is crucial. Early depth estimation methods relied on handcrafted features and traditional techniques, which struggled with complex scenes [27–29].

The arrival of deep learning improved the quality monocular depth estimation by enabling the learning of depth representations from annotated data. A significant advancement came from Eigen et al. [30], who introduced a multi-scale fusion network for depth regression. This innovation led to further improvements through classification-based approaches [31, 32], the introduction of priors [33–35], and enhanced objective functions [36, 37].

Depth Anything by Yang et al. [38] introduced a robust monocular depth estimation model using unlabeled images, excelling in depth estimation and serving as a strong foundation for downstream tasks. The authors then introduced Depth Anything V2 [39] which improves upon this by offering more precise depth predictions, supporting a wider range of applications with varied model sizes, and enhancing fine-tuning capabilities. In our work we make use of Depth Anything V2 for 3D scene and target reconstruction.

2.3 Instance Segmentation

Image instance or object segmentation is a computer vision task that distinguishes each instance of an object within an image as a separate entity. This method provides pixel-level masks for each object instance, allowing for the identification of a single or multiple objects, which is particularly useful in complex scenes with overlapping or closely packed objects.

First advancements in instance segmentation are largely driven by Mask RCNN-based methods [40]. These region-based architectures predict masks on a low resolution grid which often blurs the fine details of larger objects. Even though bottom-up approaches, which group pixels to form object masks, can produce more detailed outputs, they generally lag behind region-based methods in performance [41, 42]. TensorMask [43], a sliding-window technique, offers high-resolution masks for large objects but with slightly lower accuracy. The introduction of transformers [44] and their subsequent use in image processing [45] ushered significant improvements in instance segmentation [46–48].

Building upon the transformer architecture, Kirillov et al. introduced the Segment Anything model [49] (SAM), a promptable foundation model for image segmentation trained on a large amount of data. We use SAM for detecting the camera-radar correspondences.

3 Camera–Radar Calibration

Extrinsic calibration involves determining the transformation—comprising both rotation and translation—between the coordinate systems of different sensors. This transformation allows for the projection of points from one coordinate system to another and enables the reconstruction of 3D positions.

In radar calibration, a specific target is employed to capture and reflect the radar signal. This target, known as a retroreflector, is characterized by its ability to reflect radiation back to the source (i.e., the radar) with minimal scattering. In this study, a corner reflector is utilized, featuring a pyramidal shape composed of three right isosceles triangles joined at their vertex angle (see Figures 3 and 9).

3.1 Notation

In this section we define the notation to be used in the rest of the paper. We adopted the notation defined in [50] to describe the different relations between the coordinate systems for extrinsic calibration.

Let $\boldsymbol{m}_a = [x_a, y_a, z_a]^\top$ be a point $\boldsymbol{m}$ in a Cartesian coordinate system A. We define the transformation from system A to system B as $\boldsymbol{R}_{ba}$ and $\boldsymbol{b}_a$ respectively where $\boldsymbol{R}_{ba}$ represents the rotation from coordinate system A to B and $\boldsymbol{b}_a$ is the origin of system B represented in system A. Thus the transformation of $\boldsymbol{m}$ and its inverse can then be written as

$$\begin{aligned} \boldsymbol{m}_b &= \boldsymbol{R}_{ba}(\boldsymbol{m}_a - \boldsymbol{b}_a), \\ \boldsymbol{m}_a &= \boldsymbol{R}_{ab}(\boldsymbol{m}_b - \boldsymbol{a}_b), \end{aligned} \tag{1}$$

where $\boldsymbol{R}_{ab} = \boldsymbol{R}_{ba}^{-1} = \boldsymbol{R}_{ba}^\top$ and $\boldsymbol{a}_b = -\boldsymbol{R}_{ba}\boldsymbol{b}_a$. Thus, $\boldsymbol{m}_b$ can be expressed as

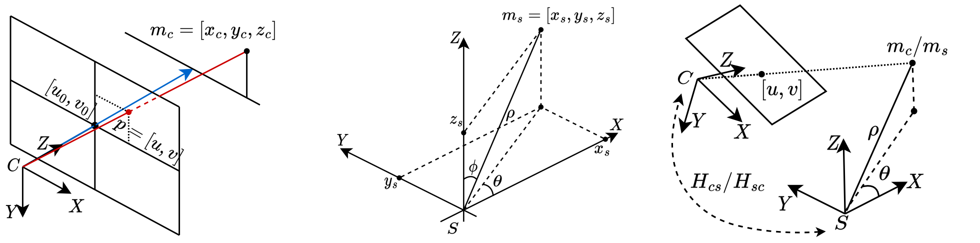
$$\boldsymbol{m}_b = \boldsymbol{R}_{ba}\boldsymbol{m}_a + \boldsymbol{a}_b. \tag{2}$$

Finally, the homogeneous transformation $\boldsymbol{H}_{ba}$ from coordinate system A to B can then be expressed as

$$\boldsymbol{H}_{ba} = \begin{bmatrix} \boldsymbol{R}_{ba} & \boldsymbol{a}_b \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{3}$$

3.2 System Model

The sensor setup consists of a radar and a camera that are rigidly connected, with a short baseline that is significantly smaller than the distance to the target being measured. To ensure clarity in terminology, we define the camera coordinate systems as C and the radar as S (sensor). Consequently, $\boldsymbol{R}_{cs}$ and $\boldsymbol{R}_{sc}$ represent the rotations



(a) (b) (c)

Fig. 2: The different coordinate systems used in defining the camera-radar calibration problem. (a) presents the pinhole model showing how objects in 3D can be described in the camera coordinate system with the pixel position on the image plane. (b) shows the radar data measured in spherical coordinates and its representation as Cartesian coordinates. (c) combines the relationships between (a) and (b) and shows how an object visible in both the camera and radar frames can be defined as well as the possible transformation between the systems. (Figure from [14])

from the radar coordinate system to the camera coordinate system and its inverse, respectively. Similarly, $\boldsymbol{c}_s$ denotes the origin of the camera in the radar coordinate system, while $\boldsymbol{s}_c$ represents the origin of the radar in the camera coordinate system.

The pinhole camera model, shown in Figure 2a, is used to project a point $\boldsymbol{m}_c = [x_c, y_c, z_c]^\top$ in the camera coordinate system onto the image plane as $\boldsymbol{p} = [u, v, 1]^\top$.

$$\begin{gathered} z_c \boldsymbol{p} = \boldsymbol{K} \boldsymbol{m}_c, \\ z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}, \end{gathered} \tag{4}$$

where $\boldsymbol{K}$ is the intrinsic parameters matrix of the camera estimated using the standard method in [51] and u and v are the pixel coordinates of the point $\boldsymbol{m}$ in an image.

The radar sensor used is a frequency-modulated continuous-wave (FMCW) radar, capable of measuring range, azimuth, Doppler velocity, and radar cross section (reflection amplitude). For calibration, we will only use the range and azimuth (ρ, θ) measurements. It is worth noting that the radar does not provide elevation information ϕ, where ϕ denotes the angle relative to the positive z-*axis*. We define the point $\boldsymbol{m}_s = [x_s, y_s, z_s]^\top$ in the radar coordinate system as shown in Figure 2b as follows

$$\begin{aligned} x_s &= \rho \sin\phi \cos\theta, \\ y_s &= \rho \sin\phi \sin\theta, \\ z_s &= \rho \cos\phi. \end{aligned} \tag{5}$$

Existing approaches [5–8, 16, 18] interpret radar data in 2D and assume that $\phi = \pi/2$ since it is usually unknown (unmeasured).

We can transform a radar point in the radar coordinate system to the camera coordinate system using

$$\begin{aligned} \boldsymbol{m}_c &= \boldsymbol{R}_{cs}\boldsymbol{m}_s + \boldsymbol{s}_c, \\ \text{or } \begin{bmatrix} \boldsymbol{m}_c \\ 1 \end{bmatrix} &= \boldsymbol{H}_{cs} \begin{bmatrix} \boldsymbol{m}_s \\ 1 \end{bmatrix}. \end{aligned} \tag{6}$$

We combine Equation (4) with Equation (6) to obtain a relationship describing the transformation between the radar and image data as shown in Figure 2c

$$z_c\boldsymbol{p} = [\boldsymbol{K} \mid \boldsymbol{0}]\boldsymbol{H}_{cs} \begin{bmatrix} \boldsymbol{m}_s \\ 1 \end{bmatrix}, \tag{7}$$

the $\boldsymbol{H}$ matrix is 4×4 (see Equation (3)), so $\boldsymbol{K}$ is extended by a zero column to match the dimensions.

3.3 Proposed Approach

Our objective is to establish a system of equations to calculate the residuals required for the optimization process. The residuals are minimized by determining the parameters for the extrinsic calibration. We begin by defining the geometric relationships that characterize the measurements, followed by formulating the optimization problem. Finally, we reconstruct the 3D point cloud using the estimated calibration parameters.

3.3.1 Geometric Constraints

By applying the measurement principles of the sensors in use, various constraints and relationships become evident. When the distance to a target is measured by the radar, the target's position is confined to a sphere with a radius of ρ, centered at the radar

$$x_s^2 + y_s^2 + z_s^2 = \rho^2. \tag{8}$$

Given the azimuth angle of the target relative to the radar's positive *x-axis*, the target lies on a plane that passes through the radar center and is perpendicular to the *xy-plane*. The normal vector to this plane is defined by the angle $(\theta + \pi/2)$. So we represent the unit normal vector to the plane passing through the target point and the radar center as

$$\begin{aligned} \overrightarrow{\boldsymbol{n}} &= (\cos{(\theta + \frac{\pi}{2})}, \sin{(\theta + \frac{\pi}{2})}, 0) \\ &= (-\sin\theta, \cos\theta, 0), \\ \text{or } \overrightarrow{\boldsymbol{n}} &= (\sin\theta, -\cos\theta, 0). \end{aligned} \tag{9}$$

We can limit the locus of the target in the radar coordinate system to the intersection between the sphere defined in Equation (8) and the plane

$$x_s \sin\theta - y_s \cos\theta = 0, x_s > 0, \tag{10}$$

the condition $x_s > 0$ ensures that the target is in front of the radar and belongs to the positive semi-circle .

Finally, the target lies on the line that passes through the camera center and the point (u, v), which is the target's projection on the image plane. This line intersects the semicircle defined by Equations (8) and (10) at a single point, corresponding to the target's position in 3D space.

3.3.2 Optimization Formulation

We setup an optimization system using Equations (8) and (10) and based on the defined constraints as follows

$$\begin{gathered} x_s^2 + y_s^2 + z_s^2 - \rho^2 = \epsilon_1, \\ x_s \sin\theta - y_s \cos\theta = \epsilon_2, \end{gathered} \tag{11}$$

such that ϵ_1 and ϵ_2 are the residuals to be minimized as to ensure the 3D radar points satisfy the constraints. We then use the position of the target in the image to derive the representation of $m_s = [x_s, y_s, z_s]^\top$ in terms of (u, v) and $\boldsymbol{H}_{sc}$ using Equation (7) as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{m}_s \\ 1 \end{bmatrix} &= \boldsymbol{H}_{cs}^{-1} \begin{bmatrix} z_c \boldsymbol{K}^{-1} \boldsymbol{p} \\ 1 \end{bmatrix} \\ &= \boldsymbol{H}_{sc} \begin{bmatrix} z_c \boldsymbol{K}^{-1} \boldsymbol{p} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}_{sc} & \boldsymbol{c}_s \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} z_c \boldsymbol{K}^{-1} \boldsymbol{p} \\ 1 \end{bmatrix}, \end{aligned} \tag{12}$$

where $\boldsymbol{R}_{sc} = \boldsymbol{R}_\gamma \boldsymbol{R}_\beta \boldsymbol{R}_\alpha$ and α, β, and γ are the rotation angles around x, y, and z respectively. Thus, the parameters to be estimated are the three rotation angles and the three translations represented by $\boldsymbol{c}_s = [x_{c_s}, y_{c_s}, z_{c_s}]^\top$.

The final unknown variable to be estimated in Equation (12) is z_c. Previous work addressed this challenge by different ways: In [19], the approach involves using at least six fixed targets and accurately measuring the distances between them to determine z_c. On the other hand, the method presented in [20] requires the ability to move the entire radar-camera system, incorporating the estimation of z_c into the optimization process. In contrast, this work introduces two new methods that estimate z_c using only a single target measured at various positions, significantly simplifying the setup.

Method 1 (Using radar range as an estimate for z_c) Given that z_c represents the depth of a target relative to the camera and considering that the radar can directly measure the

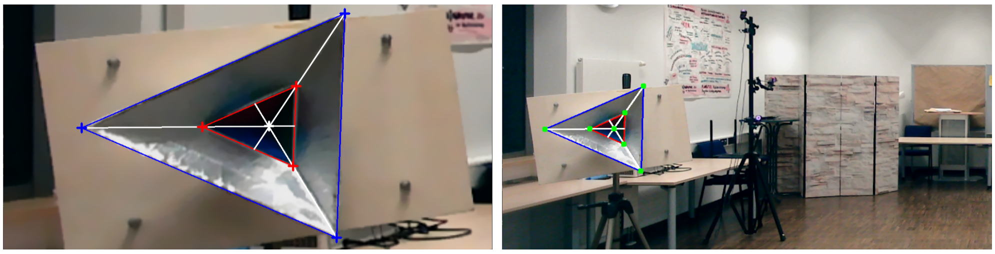


Fig. 3: (left) shows the detection of the calibration target corners, we need a minimum of 4 points to solve the PnP problem. (right) shows the reprojected solution of the PnP problem (green). (Modified from [14])

target's depth, it is reasonable to leverage the multi-modal measurement capabilities of the sensor system by setting $z_c = \rho$. This assumption holds true when the baseline between the camera and radar is significantly smaller than the distance being measured and when the camera and radar are positioned in close proximity to each other.

Method 2 (Using camera correspondences to calculate z_c) This method addresses the shortcomings of the first approach by eliminating the need for a short baseline. By utilizing the known dimensions of the radar retroreflector along with the intrinsic calibration matrix $\boldsymbol{K}$, the perspective-n-point (PnP) problem can be solved to determine the 6 *DoF* pose within the camera coordinate system [52]. The Euclidean distance to the center of the retroreflector is then considered as z_c.

The target is detected and matched to a labeled template to align the corners using the GMS Feature Matcher [53], after which the PnP problem is solved based on the aligned corners. It is important to note that limiting the search area enhances the reliability of the matching process. Figure 3 illustrates the reprojection of the reflector corners. A minimal reprojection error signifies accurate pose estimation. The matching procedure is explained in detail in Section 3.4.

For the PnP formulation, we used the iterative PnP implementation [54] on the 7 target points detected (center, and 2 on each axis). The optimization nature of the PnP solver decreases the overall error for all the points and thus reduces the effect of pixel noise or misalignment of a particular target.

3.3.3 Elevation Constraint

In addition to the residuals outlined in Equation (11), an additional residual is incorporated as a stabilizing term in the optimization process. This term helps to restrict deviations in the pitch angle β and accelerates convergence. Radar sensors typically have a relatively narrow vertical field of view (±15°), which results in data being primarily concentrated around the *xy-plane*. The stabilizing residual is formulated based on these characteristics

$$|z_s| = \epsilon_3, \tag{13}$$

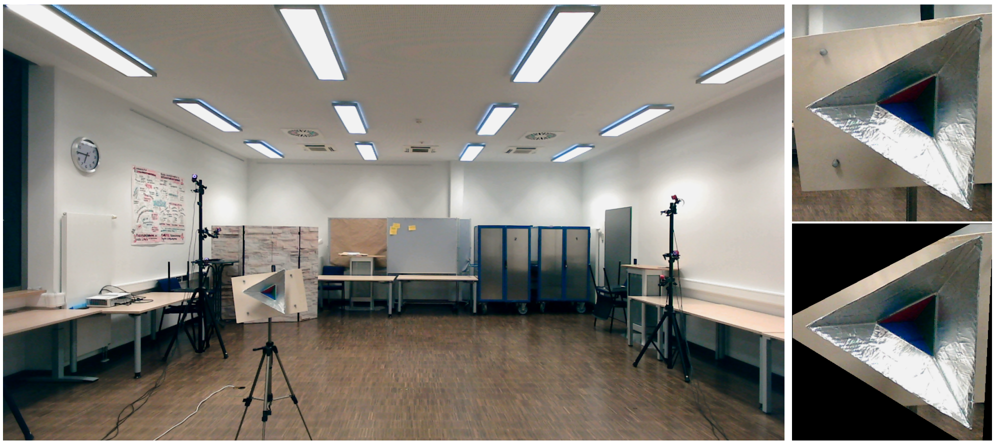
The system of optimization equations is solved using the Levenberg-Marquardt (LM) non-linear least squares optimization [55]. The goal is to find the set of parameters $[\alpha, \beta, \gamma, x_{c_s}, y_{c_s}, z_{c_s}]$ (rotation angles and translations) that minimizes the sum of squared residuals from Equations (11) and (13), $(\epsilon_1)_i^2+(\epsilon_2)_i^2+(\epsilon_3)_i^2$, for each measured target i.

Finally, we can formulate the objective function as

$$\underset{\alpha,\beta,\gamma,x_{c_s},y_{c_s},z_{c_s}}{\arg\min} \sum_{\forall i} (\epsilon_1)_i^2 + (\epsilon_2)_i^2 + (\epsilon_3)_i^2. \tag{14}$$

3.4 Target Matching in Camera Frames

In this section, we will provide a detailed explanation of the process for automatically detecting the calibration target, including the identification of its corners and center, as required by Method 2. This task is independent of the calibration algorithm and can be accomplished using various alternative methods.


(a)

(b)

Fig. 4: (a) sample frame where the corner targets must be detected. (b) patch of the calibration target and a masked version of the patch used for template matching. (Modified from [14])

3.4.1 Template Matching

Given an example input frame, as illustrated in Figure 4a, the goal is to first identify the target's location within the frame. This step is crucial to narrowing down the search space for feature matching, thereby enhancing its robustness.

To locate potential positions of the target within the input frame, a template (Figure 4b) is employed. The process involves applying template matching [56] to the

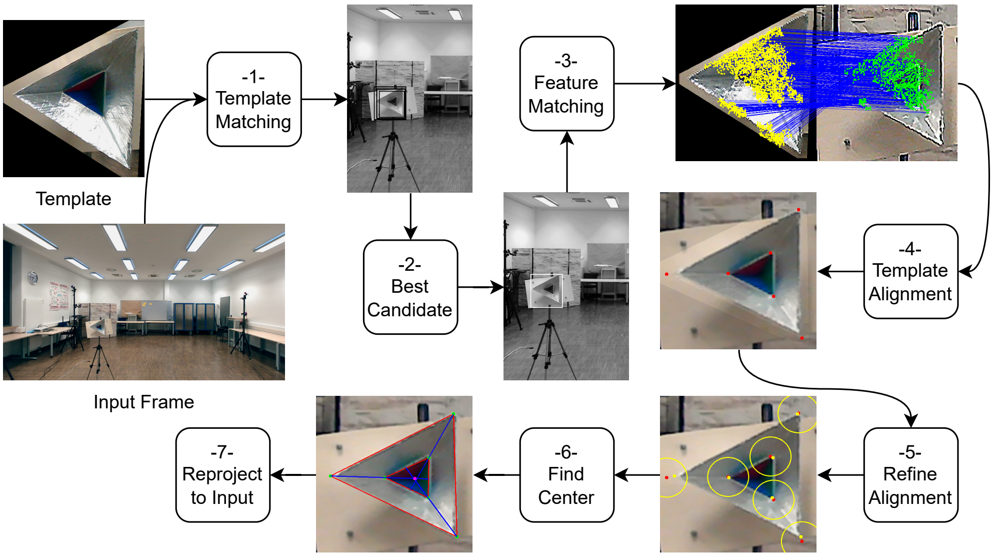


Fig. 5: The pipeline used for target detection in images: (1) generate candidate positions using image template matching. (2) combine patches to get best candidate. (3) detect matches between the candidate patch and used template using a feature matcher. (4) compute homography to align the patch and template, the red dots show the known corner positions. (5) refine corner positions using optical flow method, the yellow dots show the refined position of the corner. (6) calculate the center of the retroreflector using the intersection of the lines connecting the aligned corners. (7) project the center to the original image. (Figure from [14])

input frame across various scales and rotations of the template. Our experiments indicate that masking out the surrounding clutter in the template, as shown in Figure 4b, significantly improves detection robustness.

The template matching process generates multiple candidate positions, which are then merged based on their overlap values. Each merged patch receives a vote, corresponding to the number of candidates combined at that position. The outcomes of steps (1) and (2) in Figure 5 illustrate the candidate positions both before and after merging, as well as the final selected patch within the input frame.

3.4.2 Template Alignment

After identifying the target patch, we employ the GMS Feature Matcher [53] to establish correspondences between the template and the selected patch. The GMS Matcher enhances the robustness of feature matching between these two patches. Figure 5 illustrates the outcome of this feature matching process (3), which is conducted after resizing the patches to comparable dimensions and applying edge-enhancing filtering.

The matched features are then used to compute the homography [57], facilitating the alignment of the template with the image patch. In Step (4) of Figure 5, the aligned template is superimposed onto the image patch, with red dots indicating the known corner positions in the template. The figure highlights that some refinement is required to achieve accurate corner alignment.

3.4.3 Corner Refinement

To enhance the accuracy of the corner positions within the image patch, we employ the Lucas-Kanade method [58] for sparse optical flow. This approach assumes minimal displacement of the corner positions between consecutive images and is applied individually to each corner to achieve more reliable results. As depicted in Figure 5, the yellow markers (5) indicate the refined corner positions. The circle surrounding the initial red corner position has a radius of 50 pixels, with the image being upscaled by a factor of 3.5.

3.4.4 Center Detection

Once the corners of the target are detected, determining the center is a straightforward process. This is achieved by connecting the corners of both triangles and calculating the average intersection point of the three lines, as illustrated in Figure 5. Figure 6 displays the location of the center point as identified in the original image.

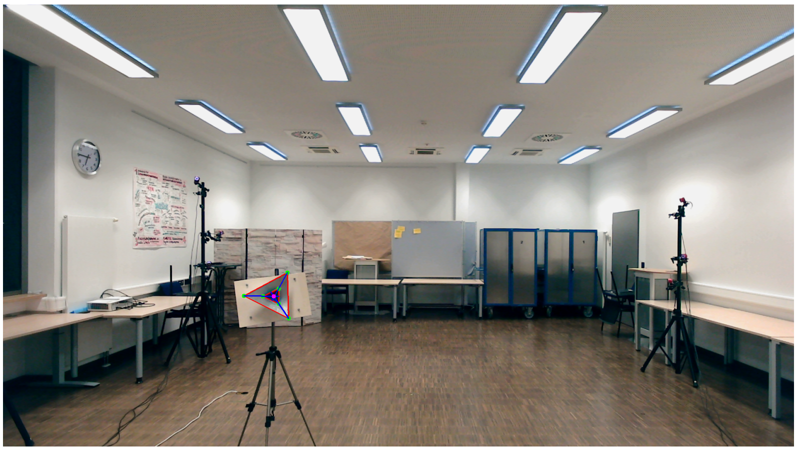


Fig. 6: The projection of the target points on the original image. (Figure from [14])

4 Point Cloud Reconstruction

In order to obtain the 3D position of radar targets and then potentially the complete point cloud of scenes, we first need to be able to perform data association between the radar and camera sensors. Unlike during the calibration when the target shape and dimensions are known and thus easy to detect in the camera as demonstrated in Section 3.4, a potential radar target can have any arbitrary shape or dimensions in other scenarios (e.g. chair, human, car)

4.1 Camera-Radar Correspondences

The task of finding the camera-radar correspondences can be described as finding the camera pixel position $\boldsymbol{p} = [u, v, 1]^\top$ corresponding to a specific radar target (ρ, θ). With the help of a pre-trained instance segmentation model such as the Segment Anything model (SAM) [49] and the extrinsic calibration computed in Section 3, it is possible to use projected radar measurement as an input prompt to the segmentation model. Finally, we average the output segmentation mask to obtain $\boldsymbol{p}$.

4.1.1 Prompt Calculation

The SAM model [49] accepts different types of prompts such as points, masks, boxes, or text to indicate to the model which objects to segment. Since the elevation of a radar measurement is unknown, it is not sufficient to assume $\phi = \pi/2$ as this can lead to an incorrect prompt, the point could be too high or too low and thus miss the correct object.

A better approach to prompting is using a box prompt and provide the model with a bigger search area for segmentation. Knowing the azimuth resolution of the radar as well as the elevation FOV, it is possible generate a meaningful bounding box. Given a radar measurement (ρ, θ), we define the corners of the prompt in the spherical coordinate system as

$$\begin{aligned} box_{min} &: (\rho, \theta - \frac{d\theta}{2}, \frac{\pi}{2} - \frac{d\phi}{2}) \text{ and} \\ box_{max} &: (\rho, \theta + \frac{d\theta}{2}, \frac{\pi}{2} + \frac{d\phi}{2}), \end{aligned} \tag{15}$$

where $d\theta$ is the azimuth resolution and $d\phi$ is the elevation resolution. Since the radar has no resolution in the elevation, $d\phi = \text{FOV}$. Using Equation (5), we can transform the limits defined in Equation (15) to the radar coordinate system and then project them on to the image by applying the extrinsic and intrinsic calibration matrices as defined in Equation (7). Figure 7 shows the resulting bounding box prompt on the image and the depth map.

4.1.2 Depth Segmentation

Monocular depth estimation models can estimate an unscaled depth map of a scene from an input image. One such model is the Depth Anything model [38, 39] which can generate high quality disparity maps.

Even though the instance segmentation can perform well when applied to the raw image, by applying the segmentation to the estimated depth or disparity[1] map we are emphasizing more the foreground objects in a scene because they are more prominent

[1] Depth and disparity are inversely related.

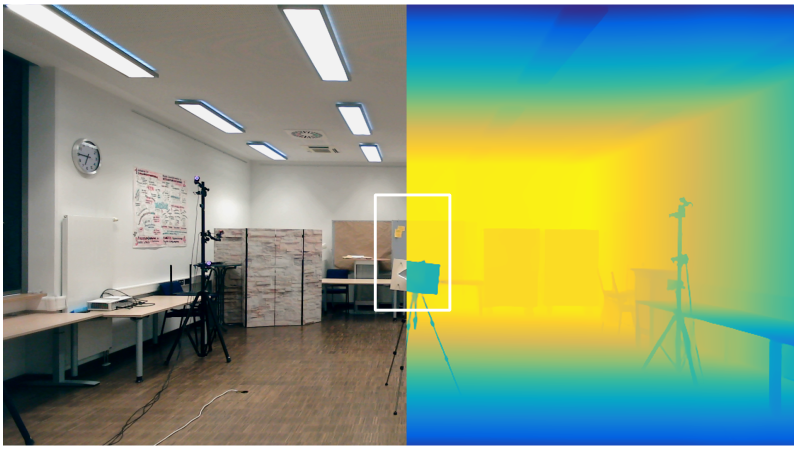


Fig. 7: The depth estimated using [39] allows the foreground objects to be more reliably segmented from the background. The white bounding box prompt area is used for instance segmentation.

in the depth map. This emphasis aligns with the assumption that the radar signals will reflect off of the foreground objects rather than penetrate them.

Figure 7 compares a generated depth map with its original input image. It shows that the depth map makes the object position and dimensions more distinguishable from the background.

4.2 Target Reconstruction

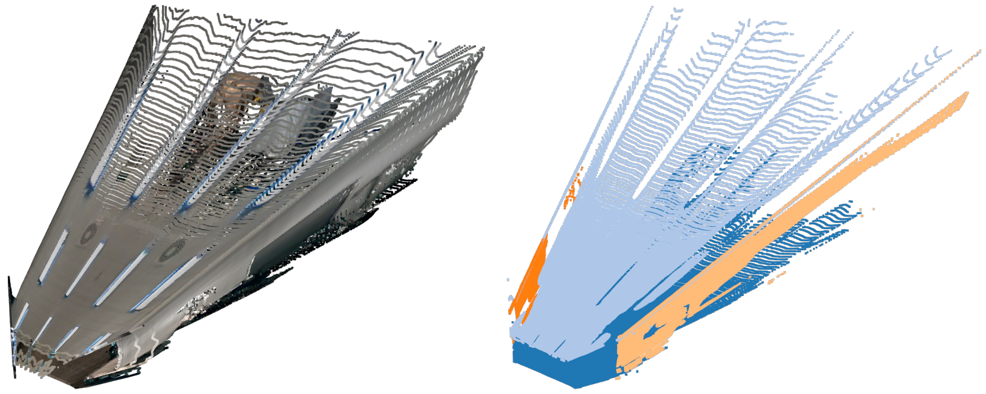
Given the camera-radar correspondences (see Sections 3.4 and 4.1), we use the computed extrinsic calibration to *fuse* the radar and camera measurements and estimate the 3D coordinates of targets, similar to [19]. Using the pinhole model in Equation (4), we represent a point $\boldsymbol{m}_c$ in terms of the image coordinates and the intrinsic calibration matrix $\boldsymbol{K}$

$$\boldsymbol{m}_c = z_c \boldsymbol{K}^{-1} \boldsymbol{p} = z_c \boldsymbol{q}, \tag{16}$$

where $\boldsymbol{q} = [q_1, q_2, q_3]^\top = \boldsymbol{K}^{-1}\boldsymbol{p}$. To compute z_c, the equation of the sphere in Equation (8) is used in the camera coordinate system and replacing the values for $\boldsymbol{m}_c$ as in Equation (16)

$$\begin{aligned} &(x_c - x_{s_c})^2 + (y_c - y_{s_c})^2 + (z_c - z_{s_c})^2 = \rho^2 \\ \Rightarrow &z_c^2(q_1^2 + q_2^2 + q_3^2) - 2z_c(q_1 x_{s_c} + q_2 y_{s_c} + q_3 z_{s_c}) \\ &+ (x_{s_c}^2 + y_{s_c}^2 + z_{s_c}^2 - \rho^2) = 0. \end{aligned} \tag{17}$$

The solution to the quadratic equation in Equation (17) yields two possible solutions. The correct solution is the one that gives a closer results of $\boldsymbol{m}_s = \boldsymbol{H}_{sc}\boldsymbol{m}_c$ to Equation (5) with $\phi = \pi/2$.



(a)

(b)

Fig. 8: (a) Original 3D point cloud reconstruction prior to optimization. (b) Planes detected using RANSAC plane fitting [60].

4.3 Scene Reconstruction

The Depth Anything V2 model estimates a disparity map of the input image based on an arbitrary baseline. A baseline is the distance between two camera centers in a stereo or multi-view setup, and the disparity is the difference in pixel positions of an object between the camera frames [59]. The equation describing the relationship between depth and disparity is

$$z_c = \frac{b \times f}{d}, \tag{18}$$

where z_c is the Cartesian depth of a point in meters, b is the baseline in meters, f is the focal length of the camera, and d is the disparity value at that point.

After obtaining the 3D position of a known point using the method described in Section 4.2, and knowing the focal length of the camera, it is possible to solve for the baseline b using Equation (18). We then use the calculated b to apply Equation (18) on each disparity value in the image to get the resulting scene point cloud. Figure 8a shows an initial reconstruction result prior to optimization.

4.3.1 Manhattan World Assumption

Using the raw disparity d output of the Depth Anything model generates an oddly shaped and elongated point cloud as seen in Figure 8a. This is an expected behavior of deep learning models since the output is usually not scaled and is influenced by the data used during training.

To obtain the new disparity d' we introduce an optimization to find the optimal $g(d)$ such that

$$\begin{aligned} g &: d + k, \text{where } k \in \mathbb{N} \\ \Rightarrow d' &= d + k, \end{aligned} \tag{19}$$

where k is a constant offset to be determined through minimizing the appropriate loss.

We base our loss on the Manhattan World assumption [61], which is a concept in computer vision that posits that many scenes, particularly urban and indoor environments, can be approximated by a 3D Cartesian grid. This means that the dominant structures in these scenes, like walls, tend to be aligned with the Cartesian axes (x, y, z). In particular, we assume that opposite walls are parallel to each other and perpendicular to the floor and ceiling.

Using the Random Sample Consensus algorithm [60] (RANSAC) for plane detection, we detect the 4 most dominant planes in the estimated point cloud: two opposite walls, ceiling, and floor seen in Figure 8b. Knowing that each plane defined by a normal vector $\overrightarrow{\boldsymbol{n}}$ and a scalar, we then define the Manhattan loss ϵ_m as

$$\epsilon_m = \frac{1}{2}\sum_{\forall i,j} \epsilon_{ij} \text{ where } \epsilon_{ij} = \begin{cases} \mid \overrightarrow{\boldsymbol{n}_i} \cdot \overrightarrow{\boldsymbol{n}_j} \mid \text{ for } \mid \overrightarrow{\boldsymbol{n}_i} \cdot \overrightarrow{\boldsymbol{n}_j} \mid < \cos 45^\circ \\ \parallel \overrightarrow{\boldsymbol{n}_i} + \overrightarrow{\boldsymbol{n}_j} \parallel \text{ for } \overrightarrow{\boldsymbol{n}_i} \cdot \overrightarrow{\boldsymbol{n}_j} < 0 \\ \parallel \overrightarrow{\boldsymbol{n}_i} - \overrightarrow{\boldsymbol{n}_j} \parallel \text{ for } \overrightarrow{\boldsymbol{n}_i} \cdot \overrightarrow{\boldsymbol{n}_j} > 0 \end{cases} \tag{20}$$

Equation (20) is minimized when the walls are perpendicular to the ceiling and floor and their normal vectors $\overrightarrow{\boldsymbol{n}_i}$ and $\overrightarrow{\boldsymbol{n}_j}$ are opposite and equal. Since $g(d)$ is a linear function, the optimal k that minimizes the Manhattan loss ϵ_m can be found through a simple linear search.

5 Calibration Evaluation

The algorithm's results are compared to the work of El Natour et al. [19], which is the only existing method for 3D camera-radar calibration specifically designed for 2D radar devices in static scenarios (i.e. calibration targets are static). In contrast to [19] that evaluated only on simulated data, we performed our evaluation on real data. In addition to that, we also use simulations, to evaluate specific aspects of the method, such as the robustness to input noise levels. The methods have been evaluated using highly accurate ground truth data obtained from an optical tracker.

5.1 Ground Truth Acquisition

The evaluation of calibration algorithms and 3D reconstruction requires a highly accurate and precise method for measuring ground truth values. To achieve this, an optical motion capture system (OptiTrack) capable of errors of less than 1 mm is utilized. It is worth noting that this system is used solely for the quantitative evaluation of the calibration results and is not used in the calibration algorithm itself.

Eighteen OptiTrack Flex 13[2] cameras are mounted to cover the area where the calibration target is positioned. Reflective markers are attached to both the sensor and the calibration targets, enabling detection by the cameras. The setup ensures that both the camera-radar system and the calibration target remain within the cameras' field of view (FOV) at all times, allowing for precise measurement of their relative positions.

[2] https://optitrack.com/cameras/flex-13/

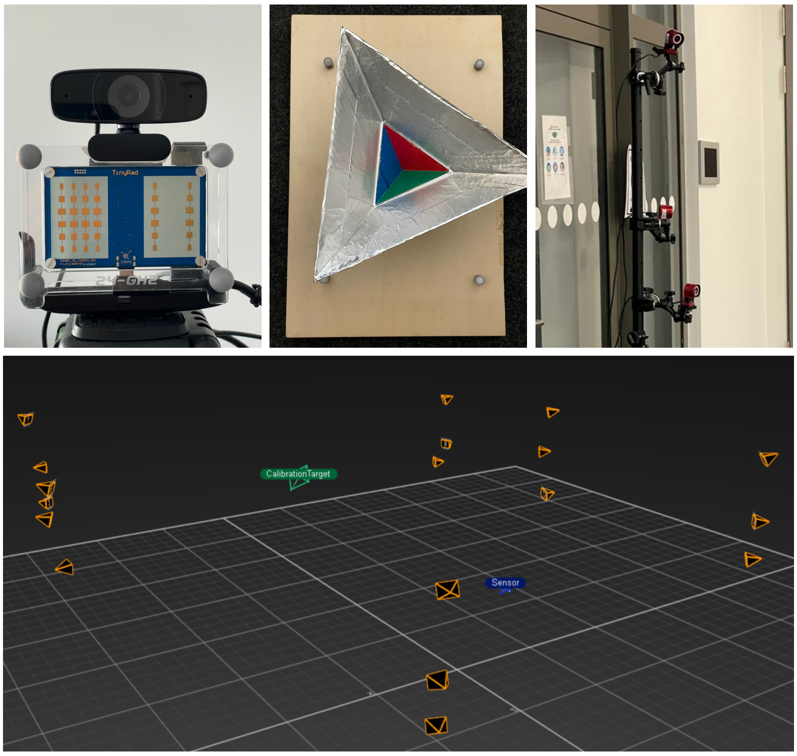


Fig. 9: (top) the camera-radar setup with the reflective markers, the calibration target, three of the OptiTrack cameras used for ground truth target pose measurement. (bottom) a 3D view showing the calibration target in green and the sensor in blue detected by the OptiTrack cameras. (Figure from [14])

As the motion capture system relies on infrared light to detect the reflective markers, calibration measurements using this ground truth method can only be conducted indoors.

To align the OptiTrack measurements with the camera-radar system, we used the reflective markers placed on the radar (as shown in Section 5.1) to calculate the radar plane and its transformation to align with the yz-plane and centered at $(0, 0, 0)$. We then used this transformation to shift all the measurements to a common reference frame with the center of the radar being its center. Since the scenes are static, no temporal synchronization was needed. The OptiTrack measurements over each static scene where averaged for each reflector to further improve the detection reliability.

5.2 Hardware Setup

The radar sensor utilized in this study is the Analog Devices TinyRad[3], an evaluation module operating at 24 GHz, offering a range resolution of 0.6 m and an azimuth resolution of 0.35 rad.

The camera used in the setup is equipped with a 2 megapixel sensor, capable of capturing full HD images (1080p) at 30 frames per second (FPS) with a field-of-view (FOV) of 78°.

The sensors are mounted with the camera positioned above the radar, separated by a short baseline of approximately 5 cm, as shown in Figure 9.

5.3 Initialization

Given the known differences in coordinate system orientations between the camera and radar, an initial parameter vector of $[\alpha_0, \beta_0, \gamma_0, 0, 0, 0]$ is used to align the axes, thereby accelerating the convergence of the calibration optimization.

While [19] relies on stereo-based 3D reconstruction to derive both their ground truth and the necessary *a priori* inter-target distance, we instead use the more precise OptiTrack data to replicate their approach. The inter-target distance is required for solving an initial optimization problem that calculates the z_c values needed to solve the primary calibration task.

Additionally, it was not feasible to reproduce and converge the z_c solution from [19] using a zero vector as the initial condition for the optimization. Since the original authors did not provide instructions for reproducing their results, the radar range ρ is used for initialization to obtain accurate estimates. Alternatively, manually measuring the distances could be considered, although this would further complicate the calibration setup.

5.4 Calibration Results

We used various criteria to assess the quality of calibration in both 3D and 2D. The optimization convergence is evaluated through different initialization parameters, and the calibration quality is evaluated in relation to the number of measurements required and the noise level in the measurements. Additionally, we performed an ablation study to highlight the significance of the elevation constraint.

In 3D, the error is defined as the distance between the estimated 3D reprojection of the target as described in Section 4.2 and the ground truth as determined by the OptiTrack system. In 2D, the error is measured as the distance between the projection of the estimated 3D target and the ground truth on the *xy-plane*.

In our previous work [14], the LM optimization was used to estimate the *offset* to the initialization vector $[\alpha_0, \beta_0, \gamma_0, 0, 0, 0]$ in order to ensure the convergence to the smallest parameter vector and avoid any non-linearities that can arise from the periodic nature of angles. After running more experiments, we found out that this had some side effects that prevented [19] to converge properly. We have then switched

[3] https://www.analog.com/en/resources/evaluation-hardware-and-software/evaluation-boards-kits/eval-tinyrad.html

to solving the minimization problem around the initialization vector instead and we present the updated results.

5.4.1 Evaluation of the Initialization

To assess the impact of initialization on calibration, we conducted optimization using various starting parameters while keeping the experimental setup consistent across all trials. We repeated the experiment a total of 250 times to achieve more statistically significant results. Table 1 demonstrates that our method achieved lower average errors and standard deviations across all initialization scenarios and significantly outperformed the competing approach for **moderate** and **bad** initializations. The initialization levels are defined as

$$\begin{aligned} \textbf{Best: } & [\alpha_0, \beta_0, \gamma_0, 0, 0, 0], \\ \textbf{Moderate: } & [\alpha_0, \beta_0, \gamma_0, 0, 0, 0] + \boldsymbol{\mu}_{1\times 6}, \\ \textbf{Bad: } & [\alpha_0, \beta_0, \gamma_0, 0, 0, 0] + \boldsymbol{\nu}_{1\times 6}, \\ \text{where } \boldsymbol{\mu}_{1:3} \in & [-1\ rad, 1\ rad] \ \ \& \ \ \boldsymbol{\mu}_{4:6} \in [-0.1, 0.1] \\ \text{and } \boldsymbol{\nu}_{1:3} \in & [-2\ rad, 2\ rad] \ \ \& \ \ \boldsymbol{\nu}_{4:6} \in [-0.5, 0.5], \end{aligned} \tag{21}$$

the components of μ and ν are uniformly sampled from the respective ranges and added to the initialization parameters described in Section 5.3.

Table 1: Mean error comparison between our method and [19] for different initialization setups to evaluate the sensitivity of the optimizations to their initialization. Where **Best:** $[\alpha_0, \beta_0, \gamma_0, 0, 0, 0]$, **Moderate:** $[\alpha_0, \beta_0, \gamma_0, 0, 0, 0] + \boldsymbol{\mu}_{1\times 6} \ni \{\boldsymbol{\mu}_{1:3} \in [-1\ rad, 1\ rad] \ \& \ \boldsymbol{\mu}_{4:6} \in [-0.1, 0.1]\}$, and **Bad:** $[\alpha_0, \beta_0, \gamma_0, 0, 0, 0] + \boldsymbol{\nu}_{1\times 6} \ni \{\boldsymbol{\nu}_{1:3} \in [-2\ rad, 2\ rad] \ \& \ \boldsymbol{\nu}_{4:6} \in [-0.5, 0.5]\}$. The errors are shown in meters. All errors are calculated with respect to the OptiTrack ground truth measurements.

Method	**Error**	**Initialization**		
		Best	**Moderate**	**Bad**
Using camera correspondences **(ours)**	3D	**0.175 *m* ±0.049**	**0.242 *m* ±0.104**	**0.346 *m* ±0.155**
	2D	**0.129 *m* ±0.065**	**0.167 *m* ±0.062**	**0.167 *m* ±0.062**
El Natour et al. [19]	3D	0.236 *m* ±0.102	0.880 *m* ±0.451	0.988 *m* ±0.478
	2D	0.131 *m* ±0.061	0.206 *m* ±0.111	0.225 *m* ±0.106

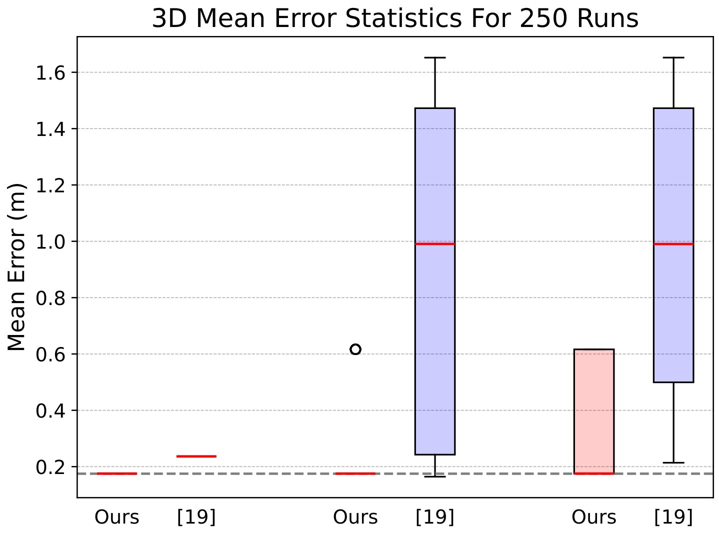


Fig. 10: Comparison of the statistics of 250 runs of the initialization experiment. From left to right, the plots correspond to the **best**, **moderate**, and **bad** initialization for our Method 2 and [19] respectively. The grey line corresponds to our method's best result from Table 1. The black circles are the outliers.

Our method consistently achieved superior results regardless of the initialization, excelling in both 3D and 2D projections on the radar and image planes. Notably, the lower standard deviation coupled with reduced error suggests greater confidence and a better fit to the data. Additionally, Figure 10 shows that our method consistently converges to the optimal solution compared to [19] for the **best** and **moderate** initialization, except for a few outliers. In case of the **bad** initialization, most of the runs converged to the optimal solution, which can be seen by the median being identical to the previous results.

5.4.2 Evaluation of the Number of Targets

Another experiment was conducted to examine how the calibration algorithms depend on the number of measurements required. The LM implementation [55] requires that the number of residuals must be at least equal to the number of parameters being estimated. As noted in Section 3.3.3, we are estimating six parameters $[\alpha, \beta, \gamma, x_{c_s}, y_{c_s}, z_{c_s}]$, with each measured target position generating three residuals. Therefore, the theoretical minimum number of target positions required is two. However, our experiments demonstrated that, in practice, three target positions are necessary to achieve convergence to a valid solution. This aligns with the broadly accepted knowledge and best practice in sensor calibration.

The results presented in Figure 11 indicate a significantly lower dependence on the number of targets for our approaches. While additional measurements offer some improvement, the calibration with just three measurements produced an error similar to [19] when using 36 measurements for 3D reconstruction. The 2D error is also lower across all experiments for our methods. This experiment was repeated 250 times for each $n \in [\![3, 36]\!]$ measurements, with measurements randomly sampled without replacement from the set of 36. The results were then averaged over all runs.

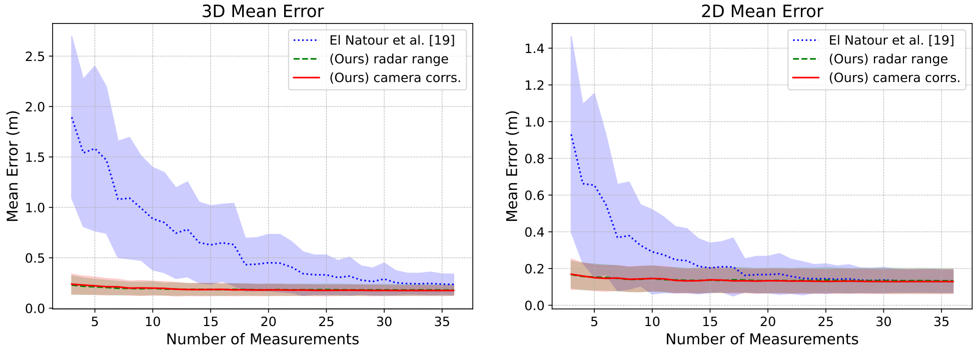


Fig. 11: Comparison showing how the number of measurements affects the calibration and the quality of the 2D and 3D reconstruction. Our methods (in red and green) show similar reconstruction error and better results even for a low number of measurements.

The poor performance of El Natour's method [19] on our data is not unexpected, considering that their real-data evaluation, as reported in [19], shows an error several orders of magnitude higher than that observed in simulated data. In their real-data evaluation, [19] reports a mean error of 0.63 m, even over a longer range and in an outdoor setting, where multi-path interference is less of a factor.

5.4.3 Simulations of Noise Levels

To evaluate the robustness of our calibration we simulated three main sources of noise: radar range measurement ρ, radar azimuth measurement θ, and camera pixel error (u, v). We use OptiTrack ground truth measurements as a baseline ($level - 0$) and simulated different levels of noise defined for each target i as

$$\begin{aligned} \rho_{il} &= \rho_{i0} + \mathcal{N}(0, (0.05 \times l)^2), \\ \theta_{il} &= \theta_{i0} + \mathcal{N}(0, (0.01 \times l)^2), \\ (u_{il}, v_{il}) &= (u_{i0} + \mathcal{N}(0, l^2), v_{i0} + \mathcal{N}(0, l^2)), \end{aligned} \tag{22}$$

where l is the noise level, $\mathcal{N}$ is the normal distribution, and $l \in [\![1, 10]\!]$, and ρ_{i0}, θ_{i0}, and (u_{i0}, v_{i0}) are the $level - 0$ measurements.

We repeated the experiment for 250 times for all noise levels, and demonstrate our methods' robustness to noise. Our methods outperforms [19] for all levels of noise in 3D reconstruction as seen in the results in Figure 12. In the case of 2D reconstruction, all three methods start off with very close reconstruction error for $level - 0$ noise, but the method by El Natour et al. [19] quickly diverges from $level - 4$ noise.

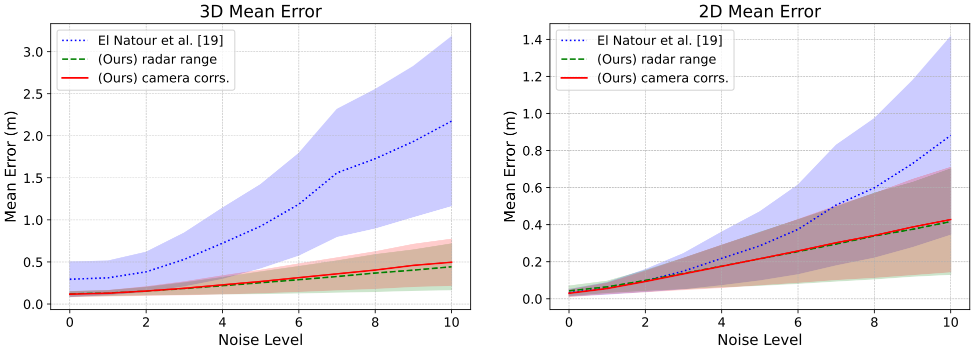


Fig. 12: Comparison showing the susceptibility to measurement noise levels. Our methods are more resilient to the increasing noise levels in 2D and 3D reconstruction compared to [19] which performs between 4 and 10 times poorer at noise level 10 in 3D reconstruction and 2D reconstruction. We achieve a worst case of 0.5 m mean error for 3D reconstruction.

5.4.4 Ablation Study of the Elevation Constraint

To prove the significance of the elevation constraint defined in Equation (13), we conducted an ablation study on both of our range calculation methods. This study was carried out using the **Best** initialization parameters outlined in Equation (21), with the results presented in Table 2. The mean errors observed when the elevation constraint was omitted were higher compared to when it was included. Furthermore, the errors without the constraint from Equation (13) closely align with the results reported in [19], as shown in Table 1.

We did another experiment without the elevation constraint but using the result of the optimization with Equation (13) as an initial condition, we show in Table 2 the results are very similar to the optimization with the elevation constraint for the camera correspondences method while performing slightly worse for the radar range method. The errors indicate that our elevation constraint allowed us to find a more optimal result. The error in the first method can be explained by the fact that it assumes the radar and camera centers to be equidistant with respect to the target.

5.5 Decoupled Noise Simulations

In addition to the experiments discussed in Section 5.4.3 regarding the combined noise levels, we conducted three additional experiments to investigate the isolated effects of the introduced noise. Similar to the procedure described in Section 5.4.3, each experiment was repeated 250 times at each noise level, with the results averaged across all runs. The noise was defined according to Equation (22), and for each experiment, noise was added to only one parameter while keeping the other two at a baseline noise level $level - 0$. Results are presented in Figure 13.

Table 2: Mean reconstruction error comparison showing the significance of adding the elevation constraint defined in Equation (13) to the optimization using both our methods. We ran the experiments with the **Best** initialization parameters $[\alpha_0, \beta_0, \gamma_0, 0, 0, 0]$ as starting condition. The third column shows the result without the elevation constraint but using the initial condition as the result of the optimization with the elevation constraint.

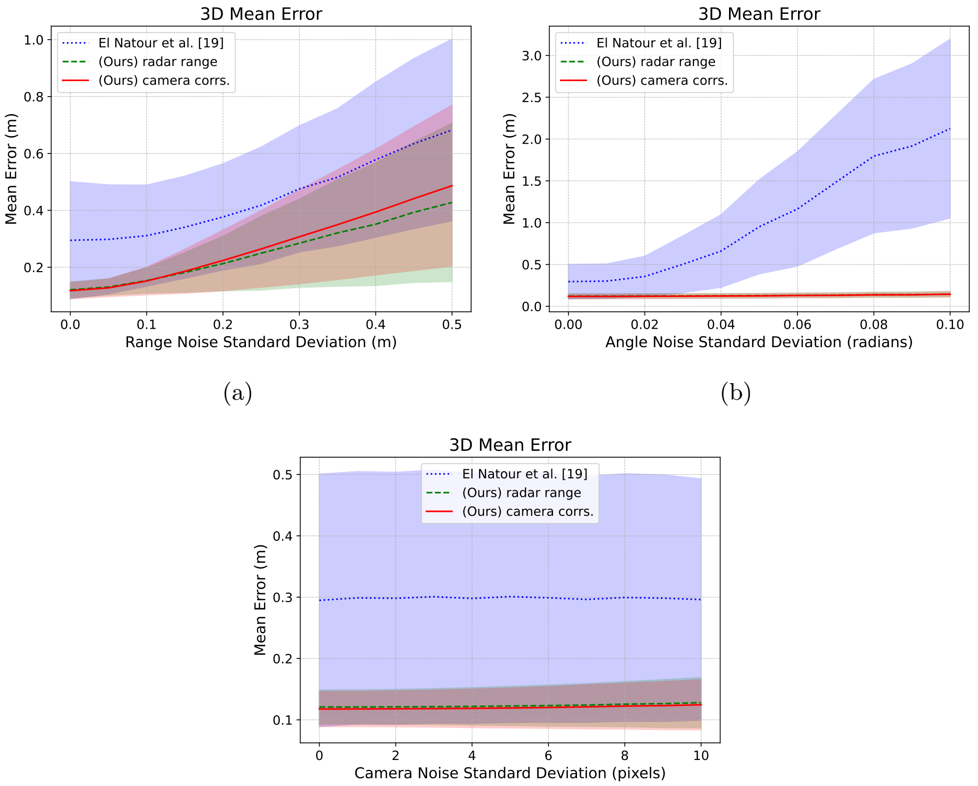
Method	**Error**	**Results**		
		Without Eq. (13)	**With Eq. (13)**	**Without Eq. (13) & optimal init.**
Using radar range **(ours)**	3D	0.262 m ±0.136	0.180 m ±0.052	0.236 m ±0.127
	2D	0.134 m ±0.065	0.133 m ±0.067	0.135 m ±0.065
Using camera correspondences **(ours)**	3D	0.223 m ±0.099	0.175 m ±0.049	**0.164 m ±0.071**
	2D	0.130 m ±0.064	**0.129 m ±0.065**	0.130 m ±0.063

Radar Range Noise – Simulations focusing on radar range noise revealed a similar trend of increased 3D mean error for both our methods and those presented by [19]. As illustrated in Figure 13a, our methods outperformed that of El Natour et al. [19] across all noise standard deviation values.

Radar Azimuth Noise – Figure 13b shows that noise in the radar azimuth measurements has the most significant impact on the performance of [19], whereas our methods showed considerably more robustness to this type of noise. Notably, the mean reconstruction error for [19] exceeded 2 m when the noise standard deviation reached 0.1 rad, while our methods maintained an error of less than 0.25 m within the same noise range.

Camera Pixel Noise – Introducing noise of up to 10 pixels had the smallest impact on all methods tested with errors remaining flat throughout the experiment.

Overall, these experiments highlight the robustness of our methods to various noise sources, with radar range noise exerting the greatest influence on our 3D reconstruction outcomes. In contrast, the approach used by [19] was notably more sensitive to variations in radar azimuth measurements and demonstrated lower overall accuracy in 3D reconstruction of the measurement targets.



(a)

(b)

(c)

Fig. 13: Decoupled noise simulation results showing 3D mean reconstruction error as a function of increased standard deviation of the noise added to (a) radar range noise, (b) radar azimuth noise, and (c) and camera pixel noise. Our methods are much less sensitive to the azimuth noise as seen in (b) while generally outperforming [19] in all three simulations.

6 Qualitative Results

In this section we evaluate our contributions to the camera-radar matching using instance segmentation as well as the 3D scene reconstruction from monocular depth estimation.

6.1 Correspondences

The correspondences detection between the camera and radar is an essential step to be able to apply the target reconstruction in an unconstrained environment, i.e. in scenarios where the shape and dimensions of the target are unknown.

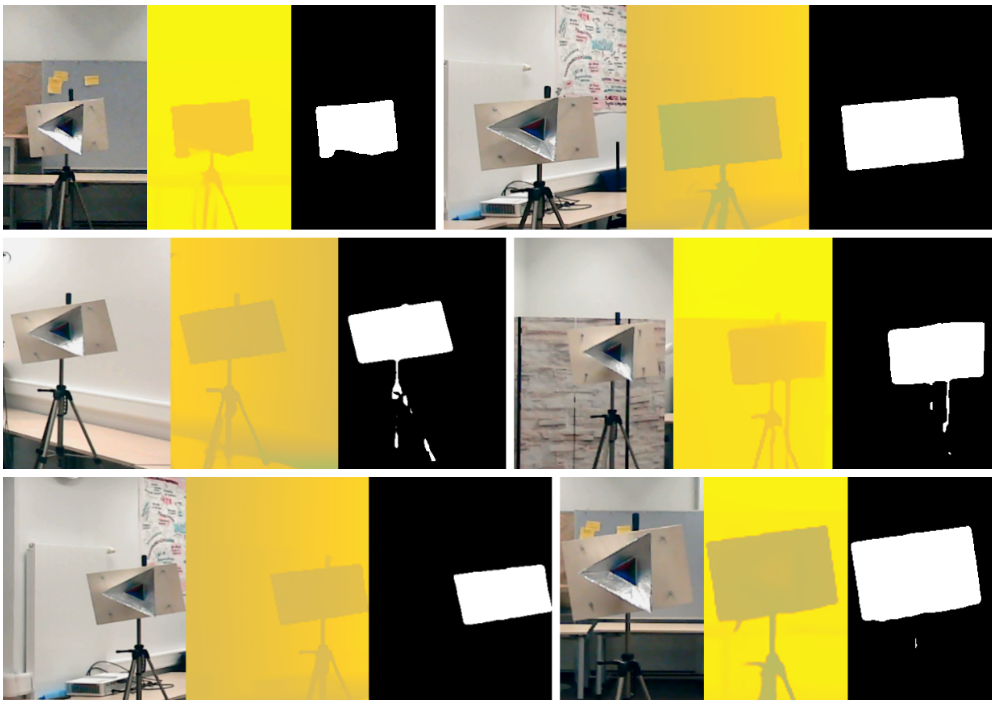


Fig. 14: Examples of the prompt bounding box calculated from the radar projection. The depth estimation shows a clear destinction between the foreground and the background and the instance segmentation results in a target mask for the camera-radar correspondences.

To localize the potential object in the image, we first estimate the depth of the image using the Depth Anything V2 model [39], then calculate the prompt area using the method described in Section 4.1.1 and Equation (15) and finally use it along with the estimated depth as inputs to the Segment Anything model [49] for instance segmentation.

Figure 14 shows some examples of the object detection in the image frame. The masks are afterwards averaged to a single pixel corresponding to the radar target. The figure also shows how the area prompt varies in size and shape depending on the detected depth and location in space. The depth estimated is applied to the full image and not only to the prompt area.

6.2 Scene Reconstruction

Following the detection of the correspondences, we reconstruct the 3D position of the target following the method detailed in Section 4.2, and then use it in combination with the estimated disparity map to generate the point cloud. To retrieve the optimized point cloud, an iterative search for the disparity shift minimizing the Manhattan loss ϵ_m is used as described in Section 4.3.1.

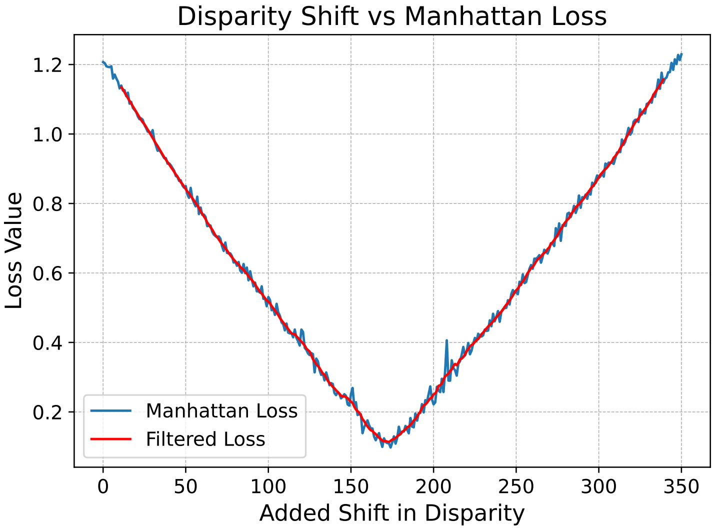


Fig. 15: Sample minimization plot of the Manhattan loss using linear parameter search, the disparity shift corresponding to the minimum loss is used in the scene reconstruction.

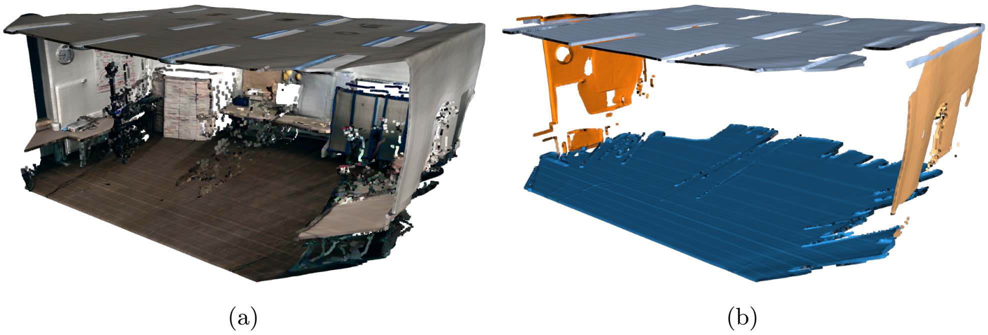
The Manhattan loss needs to be minimized for each scene separately due to the uncertainty in the depth estimation. Figure 15 shows a sample loss plot from one scene, the optimal disparity shift corresponding to the minimum loss value.

For the loss minimization we found that integer shifts to the disparity are sufficient to obtain a good quality reconstruction, however, finer search values could potentially be used. Additionally, multiple iteration can generate smoother loss curves at the expense of longer optimization times.

Following the estimation of the optimal disparity shift, we apply Equation (18) to generate the scene point cloud, Figure 8 shows the scene point cloud before optimization then Figure 16 shows the same scene after applying the Manhattan World assumption.

7 Conclusion

In this work, we introduced a new method for extrinsic calibration of a camera-radar system. This method was validated against a high-precision motion capture system, which provided the ground truth data. Our setup stands out for its simplicity, as it works independently without external sensing, while also delivering superior results. Even when starting with less accurate initial parameters and using fewer measurement points, the additional optimization constraints we implemented enable effective calibration convergence. We utilized the calibration results to reconstruct 3D targets based on data matched by the camera-radar system. Our streamlined setup, which relies on fewer calibration targets and a single standard retroreflector, eliminates


(a)

(b)

Fig. 16: (a) Optimized 3D point cloud shows a proper scene structure. (b) Planes detected in the optimized point cloud following the Manhattan assumption.

the need for more complex target designs. Although our current method is limited to static targets and scenes, incorporating a moving target during calibration could enhance radar target detection. However, this would introduce additional complexity, particularly in the setup and target detection stages of the process.

Furthermore, as this is a static calibration method, we assume a controlled environment with no interfering objects. Future research could incorporate online calibration techniques that improve the calibration over time and prevent any misalignment of sensor measurements that could occur due to physical and environmental changes.

Additionally, we implemented a camera-radar matching algorithm that utilizes pre-trained foundation models for instance segmentation and depth estimation. And combined the depth estimation with our calibration and matching results to create an end-to-end scene reconstruction pipeline capable of generating metric scale scene point clouds.

Acknowledgment

This research was partially funded by the European Union as part of the project HumanTech (Grant Agreement 101058236) and the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany as part of the research project COPPER (Grant number 01IW24009).

Declarations

Competing Interests – The authors declare no competing financial or interpersonal conficts.
Funding Information – This research was partially funded by the European Union as part of the project HumanTech (Grant Agreement 101058236) and the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany as part of the research project COPPER (Grant number 01IW24009).
Author Contribution – *Mahdi Chamseddine*: Development, results analysis, original draft, reviewing and editing. *Jason Rambach*: Supervision, cross-checking, reviewing and editing. *Didier Stricker*: Supervision, funding acquisition.
Data Availability Statement – The authors will publish the code and relevant data. github.com/mahdichamseddine/CaRaCTO.
Research Involving Human and /or Animals – Not Applicable.
Informed Consent – Not Applicable.

References

[1] Roy, A., Gale, N., Hong, L.: Fusion of doppler radar and video information for automated traffic surveillance. In: 12th International Conference on Information Fusion, pp. 1989–1996 (2009). IEEE

[2] Roy, A., Gale, N., Hong, L.: Automated traffic surveillance using fusion of doppler radar and video information. Mathematical and Computer Modelling **54**(1-2), 531–543 (2011)

[3] Cho, H., Seo, Y.-W., Kumar, B.V., Rajkumar, R.R.: A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In: International Conference on Robotics and Automation (ICRA), pp. 1836–1843 (2014). IEEE

[4] Chavez-Garcia, R.O., Aycard, O.: Multiple sensor fusion and classification for moving object detection and tracking. IEEE Transactions on Intelligent Transportation Systems **17**(2), 525–534 (2015)

[5] Sugimoto, S., Tateda, H., Takahashi, H., Okutomi, M.: Obstacle detection using millimeter-wave radar and its visualization on image sequence. In: International Conference on Pattern Recognition (ICPR), vol. 3, pp. 342–345 (2004). IEEE

[6] Wang, T., Zheng, N., Xin, J., Ma, Z.: Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications. Sensors **11**(9), 8992–9008 (2011)

[7] Wang, X., Xu, L., Sun, H., Xin, J., Zheng, N.: Bionic vision inspired on-road obstacle detection and tracking using radar and visual information. In: International Conference on Intelligent Transportation Systems (ITSC), pp. 39–44 (2014). IEEE

[8] Kim, D.Y., Jeon, M.: Data fusion of radar and image measurements for multi-object tracking via kalman filtering. Information Sciences **278**, 641–652 (2014)

[9] Stateczny, A., Kazimierski, W., Gronska-Sledz, D., Motyl, W.: The empirical application of automotive 3d radar sensor for target detection for an autonomous surface vehicle's navigation. Remote Sensing **11**(10), 1156 (2019)

[10] Wise, E., Peršić, J., Grebe, C., Petrović, I., Kelly, J.: A continuous-time approach for 3d radar-to-camera extrinsic calibration. In: International Conference on Robotics and Automation (ICRA), pp. 13164–13170 (2021). IEEE

[11] Bozic, A., Palafox, P., Thies, J., Dai, A., Nießner, M.: Transformerfusion: Monocular rgb scene reconstruction using transformers. Advances in Neural Information Processing Systems **34**, 1403–1414 (2021)

[12] Dahnert, M., Hou, J., Nießner, M., Dai, A.: Panoptic 3d scene reconstruction from a single rgb image. Advances in Neural Information Processing Systems **34**, 8282–8293 (2021)

[13] Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5511–5520 (2022)

[14] Chamseddine, M., Rambach, J., Stricker, D.: Caracto: Robust camera–radar extrinsic calibration with triple constraint optimization. In: Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - ICPRAM, pp. 534–545 (2024)

[15] Oh, J., Kim, K.-S., Park, M., Kim, S.: A comparative study on camera-radar calibration methods. In: International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 1057–1062 (2018). IEEE

[16] Domhof, J., Kooij, J.F., Gavrila, D.M.: An extrinsic calibration tool for radar, camera and lidar. In: International Conference on Robotics and Automation (ICRA), pp. 8107–8113 (2019). IEEE

[17] Peršić, J., Marković, I., Petrović, I.: Extrinsic 6dof calibration of 3d lidar and

radar. In: 2017 European Conference on Mobile Robots (ECMR), pp. 1–6 (2017). IEEE

[18] Peršić, J., Marković, I., Petrović, I.: Extrinsic 6dof calibration of a radar–lidar–camera system enhanced by radar cross section estimates evaluation. Robotics and Autonomous Systems **114**, 217–230 (2019)

[19] El Natour, G., Aider, O.A., Rouveure, R., Berry, F., Faure, P.: Radar and vision sensors calibration for outdoor 3d reconstruction. In: International Conference on Robotics and Automation (ICRA), pp. 2084–2089 (2015). IEEE

[20] El Natour, G., Ait-Aider, O., Rouveure, R., Berry, F., Faure, P.: Toward 3d reconstruction of outdoor scenes using an mmw radar and a monocular vision sensor. Sensors **15**(10), 25937–25967 (2015)

[21] Wise, E., Cheng, Q., Kelly, J.: Spatiotemporal calibration of 3-d millimetre-wavelength radar-camera pairs. IEEE Transactions on Robotics **39**(6), 4552–4566 (2023)

[22] Cheng, L., Sengupta, A., Cao, S.: 3d radar and camera co-calibration: A flexible and accurate method for target-based extrinsic calibration. In: 2023 IEEE Radar Conference (RadarConf23), pp. 1–6 (2023). IEEE

[23] Wang, Z., Zhang, L., Shen, Y., Zhou, Y.: Lvi-exc: A target-free lidar-visual-inertial extrinsic calibration framework. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 3319–3327 (2022)

[24] Wang, Y., Chao, W.-L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8445–8453 (2019)

[25] Wofk, D., Ma, F., Yang, T.-J., Karaman, S., Sze, V.: Fastdepth: Fast monocular depth estimation on embedded systems. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 6101–6108 (2019). IEEE

[26] Xu, D., Jiang, Y., Wang, P., Fan, Z., Wang, Y., Wang, Z.: Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4479–4489 (2023)

[27] Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. International Journal of Computer Vision **75**, 151–172 (2007)

[28] Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: Sift flow: Dense correspondence across different scenes. In: Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18,

2008, Proceedings, Part III 10, pp. 28–42 (2008). Springer

[29] Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE transactions on pattern analysis and machine intelligence **31**(5), 824–840 (2008)

[30] Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems **27** (2014)

[31] Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4009–4018 (2021)

[32] Li, Z., Wang, X., Liu, X., Jiang, J.: Binsformer: Revisiting adaptive bins for monocular depth estimation. IEEE Transactions on Image Processing (2024)

[33] Li, B., Shen, C., Dai, Y., Van Den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1119–1127 (2015)

[34] Yang, X., Ma, Z., Ji, Z., Ren, Z.: Gedepth: Ground embedding for monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12719–12727 (2023)

[35] Shao, S., Pei, Z., Chen, W., Chen, P.C., Li, Z.: Nddepth: Normal-distance assisted monocular depth estimation and completion. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

[36] Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., Cao, Z.: Structure-guided ranking loss for single image depth prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 611–620 (2020)

[37] Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5684–5693 (2019)

[38] Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10371–10381 (2024)

[39] Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. arXiv preprint arXiv:2406.09414 (2024)

[40] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the

IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)

[41] Liu, S., Jia, J., Fidler, S., Urtasun, R.: Sgn: Sequential grouping networks for instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3496–3504 (2017)

[42] Arnab, A., Torr, P.H.: Pixelwise instance segmentation with a dynamically instantiated network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 441–450 (2017)

[43] Chen, X., Girshick, R., He, K., Dollár, P.: Tensormask: A foundation for dense object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2061–2069 (2019)

[44] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems (2017)

[45] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al.*: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)

[46] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-toend object detection with transformers. in eccv. Springer **1**(2), 4 (2020)

[47] Chi, C., Wei, F., Hu, H.: Relationnet++: Bridging visual representations for object detection via transformer decoder. Advances in Neural Information Processing Systems **33**, 13564–13574 (2020)

[48] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2021)

[49] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., *et al.*: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4026 (2023)

[50] Rambach, J., Pagani, A., Stricker, D.: Principles of object tracking and mapping. In: Springer Handbook of Augmented Reality, pp. 53–84. Springer, Heidelberg (2021)

[51] Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence **22**(11), 1330–1334 (2000)

[52] Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the

pnp problem. International journal of computer vision **81**(2), 155–166 (2009)

[53] Bian, J., Lin, W.-Y., Matsushita, Y., Yeung, S.-K., Nguyen, T.-D., Cheng, M.-M.: Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4181–4190 (2017). IEEE

[54] Bradski, G.: OpenCV: Pose computation overview (2025). https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html

[55] Moré, J.J.: The levenberg-marquardt algorithm: implementation and theory. In: Numerical Analysis, pp. 105–116. Springer, Heidelberg (1978)

[56] Bradski, G.: OpenCV: Template Matching (2025). https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html

[57] Bradski, G.: OpenCV: Feature Matching + Homography to find Objects (2025). https://docs.opencv.org/4.x/d1/de0/tutorial_py_feature_homography.html

[58] Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), vol. 2, pp. 674–679 (1981)

[59] Bradski, G.: OpenCV: Depth Map from Stereo Images (2025). https://docs.opencv.org/4.x/dd/d53/tutorial_py_depthmap.html

[60] Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)

[61] Coughlan, J., Yuille, A.L.: The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. Advances in Neural Information Processing Systems **13** (2000)