Multi Camera Extrinsic Calibration

Online Refinement for the Rotations of Fisheye and Long Range Cameras Mounted on a Car using Natural Feature Detection in Overlapping Fields of View

Master's thesis in Complex Adaptive Systems

AUGUST BLANC

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Multi Camera Extrinsic Calibration

Online Refinement for the Rotations of Fisheye and Long Range Cameras Mounted on a Car using Natural Feature Detection in Overlapping Fields of View

AUGUST BLANC

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Multi Camera Extrinsic Calibration
Online Refinement for the Rotations of Fisheye and Long Range Cameras Mounted on a Car using Natural Feature Detection in Overlapping Fields of View
AUGUST BLANC

Industrial supervisor at Zenseact: Erik Stenborg
Supervisor: Lars Hammarstrand, Department of Electrical Engineering
Examiner: Lars Hammarstrand, Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Birds eye view image showing the surroundings of a car. The image is generated by projecting images from four fisheye cameras onto the ground plane. The brighter areas consist of superimposed images from two neighboring cameras. The car is located in the middle black rectangle of the image.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Multi Camera Extrinsic Calibration
August Blanc
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Autonomous driving has the potential of increasing traffic safety and enabling greater automation in our society. These systems rely on high precision sensor data, not the least from cameras, in order to detect and map out the environment around a vehicle. In doing this, knowledge about how the sensors are mounted is of crucial importance. This is typically calibrated in a static rig upon manufacturing a vehicle, but after driving for a while the orientation of sensors might change due to bumps and vibrations, thus requiring regular recalibration.

This thesis presents an automated refinement method for the rotations of cameras mounted on a car. There are four fisheye cameras with a wide field of view, that together can be used to form a 360° view of the surrounding ground. This is what you typically see on the display when parking a modern car. There are also three regular cameras, of which one is looking forwards and two are looking backwards but slightly to the left and right, respectively. Given images taken by the cameras at the same time, we find pixel coordinates that correspond to the same real world point for cameras with pairwise overlapping fields of view. Using this information we then optimize all camera poses simultaneously, starting from an initial pose estimate.

The performance is evaluated by two criteria. Firstly, we consider the surround view images that are created by stitching together fisheye camera images projected onto the ground plane. Looking at the overlap between cameras, the alignment of e.g. lane markings and shadows in the ground plane indicates the quality of a calibration. Secondly, we manually annotate point correspondences for each overlapping camera pair in each scenario. These are used to calculate a quantitative error metric, essentially telling us how close the pixel coordinates are to actually originate from the same real world point for the given camera poses.

The results indicate that our method achieves an accuracy comparable to that of a static calibration rig. This is a very promising result, since the time and space consuming rig calibration could possibly be switched for a calibration method that can run while the car is driving. Additionally, the fisheye camera calibration appears to achieve an accuracy comparable to other state of the art methods, while also being applicable to a wide range of driving environments and allowing for other cameras to be included in the calibration as well.

Keywords: Computer Vision, Extrinsic Calibration, Bundle Adjustment, Surround View System, Fisheye Camera, Rotation Optimization.

Acknowledgements

First of all, I want to thank Erik Stenborg, my supervisor at Zenseact, and the other team members for all useful tips and new ideas, as well as thorough feedback on the report and the project in general. You have been very helpful both in answering my questions and asking your own critical questions about the project.

Secondly, I want to thank my examiner Lars Hammarstrand for providing continuous feedback and guidance throughout the project.

Finally, thanks to Zenseact and its employees for an inspiring work environment, as well as the frequent announcements that there is leftover fika for anyone (who runs there fast enough) to enjoy.

August Blanc, Gothenburg, May 2025

Contents

List of Figures

List of Tables

1

Introduction

Autonomous systems, including robots and self driving cars, are undergoing significant development in today's society. When these systems are designed to operate reliably, they have the potential to deliver substantial benefits, particularly in enhancing safety and enabling greater automation. To function effectively they rely on sensors such as cameras and lidar, which gather real time data from the surrounding environment. This data is then processed to support decision making processes onboard the system. The environment that is measured is often complex, non homogeneous and changes with time, highlighting the need for robust collection and processing of data.

This thesis is conducted in collaboration with Zenseact, where efforts are being made to create such autonomous driving systems for cars, thus requiring high precision sensor data. One hurdle is that as the vehicles drive around, the orientation of sensors may change, causing a discrepancy between how they were mounted in the factory and their actual poses relative to the car. This rotation deviation may be on the order of several degrees, which produces visible errors when stitching overlapping fisheye camera images together. It also limits the accuracy of object detection and localization for both fisheye and regular cameras.

The current method used at Zenseact to solve the extrinsic calibration problem is based on [1]. It finds an optimal least squares solution to the hand-eye calibration problem, i.e. for one camera at a time. The camera rotation is found by estimating the motion through consecutive video frames. The method achieves an accuracy comparable to solutions applying nonlinear optimization to a similar problem formulation. However, it seems reasonable that one could increase both the accuracy and speed of the calibration by combining data from several sensors to formulate a joint optimization problem for the whole system.

This thesis aims to develop a multi sensor approach to the online extrinsic calibration. We do this by finding point correspondences between overlapping fields of view for both fisheye and longer range cameras mounted on a moving car, and then optimizing the camera poses such that the correspondences are the most likely to originate from the same world point. Our calibration method outputs the rotation of the cameras relative to the car body. The overarching goal was to develop a

fast and accurate calibration method, and that the overlapping fisheye images when stitched together produce well aligned birds eye view images.

1.1 Research Questions

The main research questions to be answered are the following.

- **How well does a multi sensor approach to fisheye camera calibration perform, in terms of accuracy and convergence speed, compared to the current online calibration method and pattern based calibration stations?**

- **How can the accuracy of the calibration be evaluated in a reliable way without the need for external measuring tools?**

- **How can data from other cameras be combined with data from the fisheye cameras to make a joint extrinsic calibration, and what is the performance of such a system?**

1.2 Limitations

The following limitations are put on the method and scope of the project.

- The project only considers online calibration methods, i.e. ones that run when the car is driving, as opposed to pattern based approaches such as placing checkerboards around the car.

- We only optimize for the rotations of cameras, as the translations are assumed to be fixed.

- We limit the calibration to refinement of a known initial estimate of the absolute camera poses.

2

Theory

The method considered in this thesis is to a large extent based on standard techniques from geometrical computer vision. These concepts are explained on a general level in the first section of this chapter. We describe how real world 3D points are projected to the image plane of a camera, and how to find the 3D point given two such corresponding projections of the same point into different camera views. Then we show how light is propagated through a fisheye camera lens, also describing the radial distortion model used in this project. After that, some state of the art models for finding point correspondences between images are presented. Then we discuss bundle adjustment, a very central concept to this project, which is about optimizing camera poses and 3D points in a scene simultaneously. In doing this one optimizes over rotation matrices for the camera poses, and since these matrices have nine elements but only three degrees of freedom, we also reference work on a more compact representation of rotations as well as their derivatives. Finally, some other literature on extrinsic calibration of a surround view system is presented.

2.1 Multiple View Geometry

The field of multiple view geometry mainly considers the geometrical properties of 3D objects being projected into 2D planes at different view points. This is usually applied to camera images of real world objects. By finding coordinates (pixels) that correspond to the same 3D point between such 2D images, one can both construct a point cloud of the 3D environment and find the poses (position and orientation) of the different view points (cameras). The results from this field can be used for lots of different computer vision tasks, for instance camera calibration, visual odometry and recreating/scanning 3D structures. The information in this section is based on the book [2], which also provides an even more detailed explanation for the central concepts of geometrical computer vision.

2.1.1 Projective Space

Consider the case in Figure 2.1. A camera placed at point $\mathbf{C}$ is pointing in the direction $(0, 0, 1)^\top$ expressed in its own coordinate system $(x, y, z)^\top$. Now consider a point $\mathbf{q}$ that lies in a plane orthogonal to the camera direction, which we for now can think of as the image taken by the camera. This image a two dimensional projection of a three dimensional space. The image is constructed by detecting incoming light rays passing through the camera center $\mathbf{C}$. This means that the point $\mathbf{q}$ in the image may in fact originate from any point $\mathbf{p}$ along the ray from $\mathbf{C}$ in the direction $\mathbf{q} - \mathbf{C}$. For this reason, the points $\mathbf{p}$ and $\mathbf{q}$ are equivalent in this projective space since they project to the same point. Now assume that the point $\mathbf{q}$ lies in the plane $z = 1$. We can get $\mathbf{q}$ from $\mathbf{p}$ by dividing by the third coordinate, i.e. if $\mathbf{p} = (x, y, z)^\top$ then $\mathbf{q} = (x_q, y_q, 1)^\top = (x/z, y/z, 1)^\top$. The point $\mathbf{p}$ is called a homogeneous coordinate of $(x_q, y_q)^\top$.

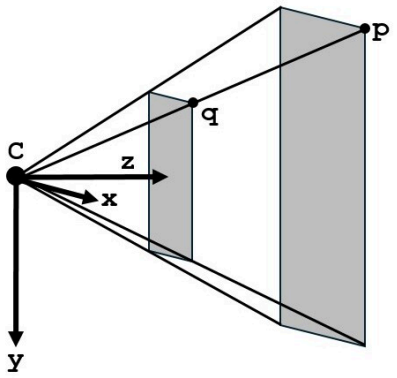


Figure 2.1: Visualization of projective space for a camera placed in C and pointing in the direction z. Images captured by the camera are 2D projections of 3D space. Thus, using a single image we only know the direction towards 3D points, but not the distance to them. Therefore the points p and q are equivalent in the projective space.

Up to this point we have only considered points in the camera coordinate system. Let's instead look at how points $\mathbf{X} = (X, Y, Z)^\top$ in some world coordinate system are projected into the camera view. To get the coordinate $\mathbf{x}$ of the point $\mathbf{X}$ transformed to the camera coordinate system we apply the camera pose matrix $P = \begin{bmatrix} R & t \end{bmatrix}$. Here $R \in \mathrm{SO}(3)$ is a rotation matrix and $t \in \mathbb{R}^3$ is a translation vector. This gives

$$\mathbf{x} \sim P \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = R\mathbf{X} + t. \tag{2.1}$$

The point is simply rotated by R and translated by t, using this notation. Also note that $\begin{bmatrix} \mathbf{X} & 1 \end{bmatrix}^\top$ is a homogeneous coordinate of the 3D point $\mathbf{X}$. Now, to find the camera pose in the world coordinate system we use the fact that the camera center $\mathbf{C}$ in the world coordinate system is the origin $\mathbf{0} = (0, 0, 0)^\top$ in the camera coordinate system, i.e.

$$\mathbf{0} = R\mathbf{C} + t \iff \mathbf{C} = -R^{-1}t = -R^\top t. \tag{2.2}$$

The rotation of the camera is the inverse of the rotation of points, i.e. $R^{-1} = R^\top$.

2.1.2 The Pinhole Camera

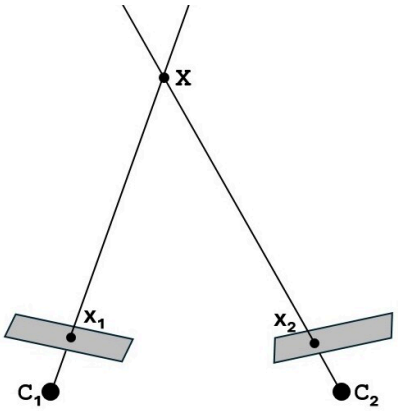
As for the actual images taken by the camera, there is a final step in the projection. In a pinhole camera incoming light rays pass through a small hole at the camera center $\mathbf{C}$ and end up on a sensor on the other side. The mapping from image plane coordinates (x, y) in the plane $z = 1$ to pixel coordinates (x_p, y_p) are given by

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = K \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tag{2.3}$$

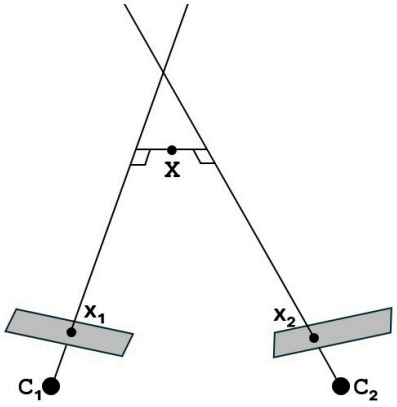
where f is the focal length and (c_x, c_y) is the principal point. These are the intrinsic (camera specific) parameters of the pinhole camera.

2.1.3 Triangulation

As we have already seen, given the projection of a 3D point $\mathbf{X}$ into one camera we can only determine the direction of the point from the camera center. However, if we can find the same point projected into a second camera, then we know that the point should lie at the intersection of the two 3D rays, as visualized in Figure 2.2a. To find such point correspondences between images, one typically uses feature extraction and matching methods, described later in this text. Because the matching process has a limited accuracy and the images have a finite resolution, the rays will in general not intersect. Instead the optimal solution is found at the closest point between the rays, as illustrated in Figure 2.2b. If we then were to project the triangulated point $\mathbf{X}$ back into the images, the reprojected point would differ from the original image points.



(a) The ideal case. The triangulated point **X** is found at the intersection between the rays.



(b) Solution with noise. The triangulated point **X** is the middle point where the rays are closest.

Figure 2.2: Triangulation of a 3D point $\mathbf{X}$ that has projected down to the image points $\mathbf{x_1}$ in camera 1 with camera center $\mathbf{C_1}$ and $\mathbf{x_2}$ in camera 2 with camera center $\mathbf{C_2}$.

Consider again the geometry of Figure 2.2b. If the distance from the cameras to the point $\mathbf{X}$ is large compared to the distance between the cameras, i.e. $||\mathbf{X} - \mathbf{C_1}|| \gg ||\mathbf{C_2} - \mathbf{C_1}||$, then a small deviation in matched pixel coordinates will result in a large uncertainty in the translation to the point. The triangulated point may even end up behind the cameras. For example, imagine that you take a picture of the sun, move a couple of meters while keeping the direction of the camera fixed, and then take another picture. The sun is very far away compared to the translation between the camera poses, so it will project to the same pixel coordinates in both images. For this reason, you can not use these images to estimate the distance to the sun. Also, you can not find the distance you moved between the images only using this information.

2.2 Fisheye Camera Model

Fisheye cameras are characterized by a wide field of view (usually around 180^o), and consequently their distortion can not be modeled accurately using a pinhole camera model. In this project, the method described in [3] is used for intrinsic calibration of both fisheye lens and long range cameras. Since the wide field of view can not be projected onto a finite image plane, we instead project the points onto a unit sphere around the camera center. The geometry is visualized in Figure 2.3. For a 3D point $\mathbf{X}$ we get the unit sphere coordinate $\mathbf{x}_{\text{sphere}} = (x_{\text{sphere}}, y_{\text{sphere}}, z_{\text{sphere}})^\top$ by

$$\mathbf{x}_{\text{sphere}} = \lambda P \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}, \tag{2.4}$$

where $P = \begin{bmatrix} R & t \end{bmatrix}$ is the camera pose matrix and λ is a constant, and then normalize by finding λ such that $||\mathbf{x}_{\text{sphere}}|| = 1$. From this we can calculate the radius $r = \sqrt{x_{\text{sphere}}^2 + y_{\text{sphere}}^2}$ and the inbound angle $\theta = \arctan(\frac{r}{z_{\text{sphere}}})$. Now we apply the radial distortion model from [3] as

$$\rho(\theta) = \theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9, \tag{2.5}$$

where $k_1, \ldots, k_4$ are intrinsic parameters. The sphere coordinates are scaled by the factor $\frac{\rho}{r}$ and then rescaled by the focal length and principal point, yielding the pixel coordinates $(x_{\text{pxl}}, y_{\text{pxl}})$ as

$$\begin{cases} x_{\text{pxl}} = f_x \dfrac{\rho}{r} x + c_x \\ y_{\text{pxl}} = f_y \dfrac{\rho}{r} y + c_y \end{cases} \tag{2.6}$$

in the captured image.

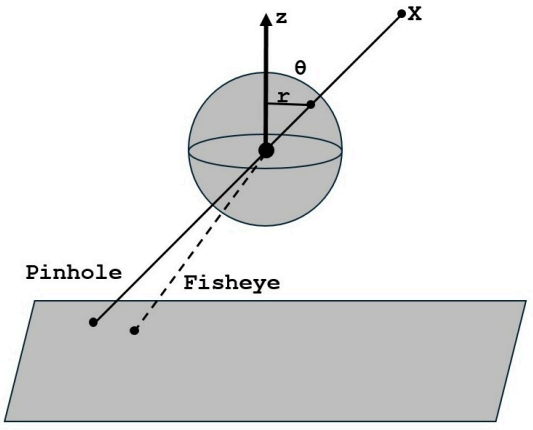


Figure 2.3: Radial distortion of the fisheye camera lens. In the fisheye camera, incoming light rays bend towards the middle of the camera sensor, where larger angles θ from the camera direction result in higher curvature. A pinhole camera, on the other hand, does not change the angle of incoming light.

2.3 Feature Extraction and Matching

Finding performant ways to match corresponding pixels between images is an active area of research, and there are a lot of different methods proposed in the literature. The process is usually divided into two steps, feature extraction and feature matching. The former is about extracting pixel coordinates and descriptors (a vector representation for the appearance of each keypoint) in individual images. The latter is about matching corresponding keypoints between images. These used to be purely algorithmic, such as the (still) widely used method SIFT [4], but nowadays practically all research in the area is focused towards machine learning approaches.

A recently developed feature extractor is ALIKE [5] and its improved version ALIKED [6]. Like previous methods it applies a convolutional neural network to the input image, which outputs a score map and a descriptor map. The score map contains a score for each pixel, representing the probability of the pixel being a keypoint. The descriptor map contains a descriptor vector for each pixel. Unlike other methods that simply choose the pixel with the highest score in regions of interest, ALIKE/ALIKED makes use of the score distribution in local score patches. For such patches, the keypoint position is calculated as the sum of pixel positions weighted by the softmax value of their score. In this way, the method is able to find features at sub-pixel positions. It achieved similar or better accuracy compared to other state of the art models, while being able to process several times as many frames per second.

When it comes to high performance feature matching, there are a lot of different methods, but one commonality is that they all seem to be based on machine learning models. Some recent ones are LightGlue [7], DiffGlue [8] and OmniGlue [9]. The accuracy is quite similar for all of these models, evaluated over general computer vision tasks.

2.4 Bundle Adjustment

As we saw in section 2.1.3 we can estimate the position of 3D points in the world by triangulating point correspondences. We also stated that the rays originating from the cameras in general do not intersect perfectly. Because of this, the projection of a triangulated point back into the camera images will end up offset from the point correspondences. A homogeneous 3D point $\mathbf{X} = (X, Y, Z, W)^\top$ projected into a camera $P = \begin{bmatrix} R & t \end{bmatrix}$ with rows P^k will end up at the image plane ($z = 1$) coordinates

$$\left(\frac{P^1\mathbf{X}}{P^3\mathbf{X}}, \frac{P^2\mathbf{X}}{P^3\mathbf{X}} \right). \tag{2.7}$$

The distance between correspondences $\mathbf{x} = (x, y)$ in image plane coordinates and reprojected points is called the reprojection error [2], which we can write as

$$\left\| \left(x - \frac{P^1\mathbf{X}}{P^3\mathbf{X}}, y - \frac{P^2\mathbf{X}}{P^3\mathbf{X}} \right) \right\|. \tag{2.8}$$

The total squared reprojection error for a system of cameras $i = 1, \ldots, m$ and 3D points $j = 1, \ldots, n$ can be expressed as the sum

$$\sum_{i,j} \left\| \left(x_{ij} - \frac{P_i^1\mathbf{X_j}}{P_i^3\mathbf{X_j}}, y_{ij} - \frac{P_i^2\mathbf{X_j}}{P_i^3\mathbf{X_j}} \right) \right\|^2 \tag{2.9}$$

of the individual errors. Bundle adjustment is the process of minimizing this entity with respect to the camera poses and 3D points (and sometimes also intrinsic parameters) simultaneously, i.e. solving the problem

$$\min_{\{P_i\},\{\mathbf{X_j}\}} \sum_{i,j} \left\| \left(x_{ij} - \frac{P_i^1\mathbf{X_j}}{P_i^3\mathbf{X_j}}, y_{ij} - \frac{P_i^2\mathbf{X_j}}{P_i^3\mathbf{X_j}} \right) \right\|^2. \tag{2.10}$$

If the error follows a Gaussian distribution with zero mean, then this minimization problem is equivalent to finding the maximum likelihood estimate [2]. Thus we can find optimal camera poses and 3D points by solving a sparse, nonlinear least squares optimization problem.

2.5 Rotation Representations

Rotation matrices R in three dimensions can be defined by

$$\{R \in \mathbf{R}^{3\times 3} \,|\, RR^\top = I,\, \det(R) = 1\}, \tag{2.11}$$

i.e. orthonormal 3×3 matrices with determinant $+1$ [10]. This also means that they have only three degrees of freedom, as opposed to the 9 matrix elements. Thus, when optimizing directly over a rotation matrix there are additional constraints that need to be enforced. Therefore it is common to use a different representation that allows for unconstrained optimization.

Another way to represent rotations is through axis-angle parametrization, i.e. a unit vector $\overline{\mathbf{v}} \in \mathbf{R}^3$ that specifies the direction of the rotation axis, and rotation of an angle θ around that axis [11]. Since $\overline{\mathbf{v}}$ is a unit vector it can be represented by two numbers, and thus there is still only three degrees of freedom. However, an even more compact parametrization is exponential coordinates given by the vector $\mathbf{v} = \theta\overline{\mathbf{v}} = \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^\top$. This relates to the rotation matrix R by the exponential map

$$R = \exp([\mathbf{v}]_\times), \tag{2.12}$$

where

$$[\mathbf{v}]_\times = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}. \tag{2.13}$$

On the contrary, to convert R to axis-angle or exponential coordinates, we have

$$\theta = ||\mathbf{v}|| = \arccos\left(\frac{\text{trace}(R) - 1}{2}\right) \tag{2.14}$$

and provided that $\theta \neq 0$

$$\overline{\mathbf{v}} = \frac{\mathbf{v}}{||\mathbf{v}||} = \frac{1}{2\sin\theta}\begin{pmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{pmatrix}. \tag{2.15}$$

In [11], an expression for the derivative of a rotation matrix expressed in exponential coordinates is derived, namely

$$\frac{\partial \mathbf{R}(\mathbf{v})\mathbf{u}}{\partial \mathbf{v}} = -R[\mathbf{u}]_\times \frac{\mathbf{v}\mathbf{v}^\top + (R^\top - I)[\mathbf{v}]_\times}{||\mathbf{v}||^2}. \tag{2.16}$$

Using these results, one may directly use the three values of $\mathbf{v}$ for the state vector to be optimized, providing a compact formulation for unconstrained optimization.

2.6 Surround View System Calibration

There exists some literature specifically on extrinsic calibration of a surround-view system, i.e. a system of fisheye cameras mounted on a vehicle. In this project only online calibration methods will be considered, eliminating pattern-based methods (like placing chessboards in overlapping camera views). A common online method is to minimize the photometric error, meaning the intensity difference in the overlap between images projected down on the ground plane. This is a reasonable method in the sense that the application of the surround view system often is to produce birds eye view images of the car surroundings. The photometric-based system in [12] achieves good short range accuracy. However, it lacks in long-range accuracy because of a limited pixel resolution in the ground plane [13], and for the purposes of this project it would also be hard to combine the method with the long range cameras that detect points further away from the vehicle. Because of this, it seems more advantageous to develop a feature-based method, i.e. detecting points of interest in the environment and either tracking them through time or finding correspondences between images from adjacent cameras with overlapping fields of view.

One method that uses only feature correspondences from the overlap in fisheye camera images is described in [14]. The article was published in 2014, before the widespread usage of neural networks for feature matching purposes. Therefore feature matching in the high distortion fisheye images was an issue. They were able to match features from objects far away, but had to project images to virtual views, specifically in the ground plane, for features closer to the cameras. Thus the geometry is approximated by a flat ground plane for close points and infinitely far away points for all other features. Initial pose estimates were refined to get relative pose measurements. Then they establish a common coordinate system to form estimated absolute poses, and then refine these estimates. The results show a somewhat good accuracy with a mean value of 0.55° error between the optimized poses and their ground truth values. The method does however seem limited by the assumptions about flat ground plane and points at infinity. Also, projected camera views typically require some interpolation between missing pixels in the new view, which might have resulted in larger errors in the feature matching process.

There are some other articles using feature based methods, but they are typically limited to special environments or require further information. For example, [15] and [16] also use odometry data and do not make use of overlapping fields of view. Another example is [17], in which the method is based on detecting lane markings in fisheye images. The front and back fisheye camera poses are estimated first, followed by the left and right camera poses, and then refinement of the estimates. They also assume the lane markings to be parallel. This results in a quite general method, and shows relatively good results, but it is still limited by the need for clear lane markings.

Recently, the article [13] was published, proposing a method where the user is required to click on matching keypoints on the ground plane in images from adjacent

cameras. It also takes as input the intrinsic calibration as well as an initial estimate of the translation and rotation of the cameras. The optimization differs from regular bundle adjustment, because the correspondences are projected to the (assumed flat) ground plane, resulting in two 3D points in this plane for each matched pixel pair. The objective function is then taken as the sum over distances between these points. The method showed a high accuracy calibration on both short and long distances in the birds eye view images. This indicates that using point correspondences in overlapping fields of view can yield high performance calibrations.

The article [13] also highlights some limitations of using birds eye view photometric error as an evaluation metric. Photometric error does not account for differences in illuminations and exposure time for the cameras. Also, points that lie above the ground plane are projected differently for different camera angles. This yields errors dependent on the specific appearance of the three dimensional structures, that should ideally not affect the outcome of the calibration. Instead the article proposes the use of mean distance error, i.e. the average distance between pixels projected back to the ground plane, and uses this for evaluation with some annotated points. Overall the method seems interesting, but has limited applicability since it requires manual clicking and only considers points in the ground plane.

3

Method

The method is in large focused towards applying local optimization to the camera poses in order to minimize the reprojection error for point correspondences in the overlapping fields of view. The chapter starts with an overview of the camera setup and data collection, and then describes the developed pipeline from start to end. The first step is about finding point correspondences between images from the camera pairs. These are undistorted to give the directions of the light rays relative to the cameras. After that the undistorted points are triangulated using initial pose estimates, and points that are too far away from the cameras are rejected. The analytical expressions needed to calculate the residual (reprojection errors) and Jacobian are derived, and are then implemented to run least squares optimization. We minimize the reprojection error with respect to the camera rotations and the position of the 3D points, starting from the CAD model rotations and the triangulated points. The optimization is performed in several stages, each one lowering the threshold of allowed residual lengths. Finally, the evaluation metrics are discussed. Both the birds eye view images and reprojection errors of manually annotated points are considered in order to enable both a qualitative and quantitative analysis.

3.1 Camera Setup

The setup consists of seven cameras of three different types mounted on a car. The camera types are listed in Table 3.1 and a top view of the setup is shown in Figure 3.1. For the optimization that is described later, the camera pairs with large enough overlap in their fields of view are used. For the fisheye cameras, this means each camera is paired with its two neighbors. The long range back left and long range back right cameras both overlap with the fisheye back camera, as well as the left and right one, respectively. The fisheye cameras are time synchronized, meaning that they capture images at the same time, whereas the long range cameras are not. This is a problem for our method since the cameras will have moved by a certain distance in the driving direction, creating a larger uncertainty in camera translations. We try to mitigate this by choosing scenarios where the car moves relatively slowly.

Table 3.1: The resolution and field of view (in the horizontal direction) for the different camera types.

Camera type	Resolution	Field of view
Fisheye	1936×1220	195°
Long range front	3848×2168	120°
Long range back	3848×2168	70°

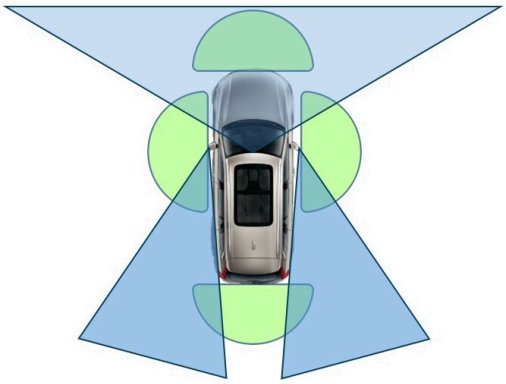


Figure 3.1: The camera setup on the cars, visualizing the approximate mounting position, direction and field of view for each camera. Fisheye cameras are depicted in green and long range cameras in blue. There are four fisheye cameras mounted in the grille, trunk lid and the left and right side view mirrors of the car. A forward looking long range camera is mounted close to the rear view mirror. There are also two long range cameras mounted in the side view mirrors looking backwards but slightly to the left and right, respectively.

3.2 Input Data

The input data consists of three parts, namely the intrinsic calibrations, the ideal poses and images captured by the cameras. The intrinsic parameters are the focal length, principal point and the distortion coefficients from (2.5). The ideal poses are given by the CAD model of the vehicle, and consist of the translation and rotation of the cameras in the vehicle coordinate frame. There are a number of images from each camera taken at the same timestep (or the closest timestep for the long range cameras). The method is quite general when it comes to driving environment, but we require that there are enough objects close to the car (within 20 m) to extract features from. For the evaluation step we also require some images taken where the ground is relatively flat and contains some type of lines (e.g. lane markings or shadows) to inspect the overlap in the birds eye view images.

3.3 Extracting Feature Correspondences

In order to find pixel coordinates corresponding to the same point, ALIKED [6] is used to extract features and LightGlue [7] is used to match the features between images. The matching is performed directly on the images, even for the fisheye cameras, as this has seemed to produce good results. When running the matching on rectified (undistorted) images, we seem to get similar results, but to some extent the fisheye images seem more natural looking. Also, the rectified image requires interpolation between some pixels, whereas some pixels are not transferred to the new image, so we lose some information in the process. A comparison is shown in Figure 3.2.

(a) Fisheye image. **(b)** Rectified image. **(c)** Image from long range forward camera.

Figure 3.2: Example of an image from the forward looking fisheye camera, the same image rectified and an image taken by the forward looking long range camera. The rectification process reverts the radial distortion, which can for example be seen by the road sign post on the right being straightened out. Note that the rectified image contains a larger field of view than the long range forward image, but in a zoomed in view of the rectified one they look very similar.

The matching is done for all overlapping camera pairs with images taken at the same timestep (or the closest available timestep in the case of long range cameras). Around 10−30 images are used for each camera, but one could of course use more if needed. We simply need enough matches to get a robust estimate of the poses in the end, after discarding bad correspondences (as explained later). We assume that

the camera poses are constant relative to the car body for the duration of the data collection. The problem is formulated in the vehicle coordinate system, so we can treat all matches as if they were from the same image. Note that the point cloud of 3D points we then get as a byproduct of the method is not a valid reconstruction of the outside world though, since the car moves between the times that images were captured.

From the matching process we get a number of pixel coordinate pairs for each camera pair. However, not all of these are true matches, as can be seen in Figure 3.3. Note that including false matches in the optimization negatively affects the accuracy of the method, as they might yield a huge reprojection error (which is the quantity to be minimized later). Now, for the fisheye cameras we would like to avoid having 3D points behind the camera at any point, so we filter out matches that have an inbound angle larger than 75°, i.e. keep only the ones within a 150° field of view. Also, closer to the edges of the fisheye images a one pixel difference corresponds to a larger angle difference in ray direction than for pixels close to the center.

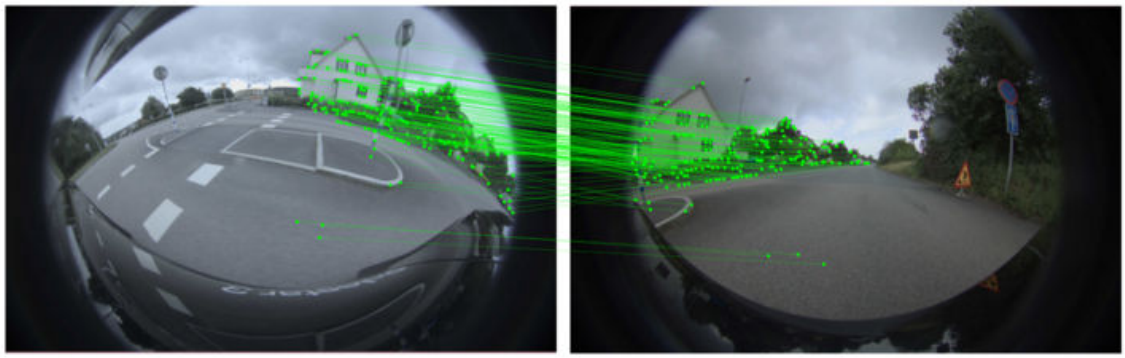


Figure 3.3: Example of matched feature coordinates between the left and front looking fisheye cameras. The feature points (green dots) are connected by the green lines to their respective matched point. Some matches are false correspondences, for example the three lowest points on the ground.

3.4 Triangulation and Filtering of Points

Now that we have found some matches, we would like to undistort the points in order to get the direction of the light rays relative to the camera poses. We can rewrite the distortion model in (2.5) as

$$\theta + k_1\theta^3 + k_2\theta^5 + k_3\theta^7 + k_4\theta^9 - \rho(\theta) = 0, \tag{3.1}$$

and then given ρ find θ from the roots of the polynomial. The undistorted matches can then be triangulated in order to give an initial estimate of the 3D points. For this the camera poses from the CAD model of the sensor setup is used.

Before starting to optimize, we do some additional filtering. Firstly, points that triangulate to behind any of the cameras are removed. Secondly, recall that the

translations of cameras do not affect the reprojection of 3D points far away from them. The translations enforce that the rotations will also be given in the vehicle coordinate system. Thus, for the case of all points being far away from the cameras, the rotations may end up all offset in some direction. The relative rotations would still be the optimal ones, but the absolute rotations (in the car coordinate system) might be wrong. Therefore, we save only the points that triangulate to within a distance of 20 m.

3.5 Local Optimization of Camera Rotations

The optimization problem can be formulated as the least squares problem

$$\min_{\mathbf{q}} \sum_{i,j \in S} ||\mathbf{r}_{ij}(\mathbf{q})||^2 \tag{3.2}$$

for cameras $i = 1, \ldots, m$ and 3D points $j = 1, \ldots, n$, where the residual $\mathbf{r}_{ij}(\mathbf{q})$ is the vector between the matched pixel coordinates and the projection of the 3D point back into the image. $\mathbf{q}$ is the state vector. Note that the sum only includes the index pairs (i, j) that are in the set S of point correspondences. Since each point j originates from the triangulation between two cameras, the sum consists of $2n$ elements. The problem can be rewritten as

$$\min_{\mathbf{q}} ||\mathbf{r}(\mathbf{q})||^2, \tag{3.3}$$

where $\mathbf{r}(\mathbf{q}) \in \mathbb{R}^{4n}$ is a single column vector of the stacked individual residuals $\mathbf{r}_{ij}(\mathbf{q})$.

3.5.1 Residual Calculation

The residual is formulated as the reprojection error

$$\mathbf{r}_{ij} = \begin{pmatrix} r_{ij,x} \\ r_{ij,y} \end{pmatrix} = \begin{pmatrix} x_{ij,\text{match}} - x_{ij,\text{reprojected}} \\ y_{ij,\text{match}} - y_{ij,\text{reprojected}} \end{pmatrix} \tag{3.4}$$

for matched pixel coordinates $(x_{ij,\text{match}}, y_{ij,\text{match}})$. $(x_{ij,\text{reprojected}}, y_{ij,\text{reprojected}})$ is the reprojection of the homogeneous point

$$\mathbf{X}_j = \begin{pmatrix} X_j^{\text{world}} \\ Y_j^{\text{world}} \\ Z_j^{\text{world}} \\ W_j^{\text{world}} \end{pmatrix} \tag{3.5}$$

in the world coordinate system into camera i. For a certain index pair i and j the reprojected point is given by

$$\begin{cases} x_{\text{reprojected}} = f_x(x_{\text{sphere}} \frac{\rho}{r}) + c_x \\ y_{\text{reprojected}} = f_y(y_{\text{sphere}} \frac{\rho}{r}) + c_y \end{cases}, \tag{3.6}$$

where $(x_{\text{sphere}}, y_{\text{sphere}}, z_{\text{sphere}})$ are the coordinates on the unit sphere around the camera center for the point $\mathbf{X}$ projected into the camera coordinate system. r is the radius

$$r = \sqrt{x_{\text{sphere}}^2 + y_{\text{sphere}}^2} \tag{3.7}$$

in the xy-plane of the camera and $\rho = \rho(\theta)$ is given by the distortion model in (2.5). In turn, θ is given by $\tan(\theta) = \frac{r}{z_{\text{sphere}}}$, and provided that the point is in front of the camera ($-\pi/2 < \theta < \pi/2$) we can apply the inverse to get $\theta = \arctan(\frac{r}{z})$.

Now denote the point $\mathbf{X}$ (in world coordinates) projected into the camera coordinate system by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = P\mathbf{X} = \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{X} = \begin{pmatrix} P^1\mathbf{X} \\ P^2\mathbf{X} \\ P^3\mathbf{X} \end{pmatrix} \tag{3.8}$$

for rows P^k of the camera matrix P. We also know that

$$\begin{pmatrix} x_{\text{sphere}} \\ y_{\text{sphere}} \\ z_{\text{sphere}} \end{pmatrix} = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \tag{3.9}$$

Then we can rewrite the expressions

$$r = \sqrt{\left(\frac{X}{\sqrt{X^2 + Y^2 + Z^2}}\right)^2 + \left(\frac{Y}{\sqrt{X^2 + Y^2 + Z^2}}\right)^2} = \frac{\sqrt{X^2 + Y^2}}{\sqrt{X^2 + Y^2 + Z^2}} \tag{3.10}$$

and

$$\theta = \arctan\left(\frac{r}{z}\right) = \arctan\left(\frac{\sqrt{X^2 + Y^2}}{Z}\right). \tag{3.11}$$

Now we have all of the expressions needed to calculate the reprojection error for a given point $\mathbf{X_j}$ and camera pose P_i.

3.5.2 Jacobian calculation

Returning to the residual expression in (3.4), we would like to calculate the Jacobian $\mathbf{J}$ as a function of the rotations $\mathbf{v}_i = (v_{i,1}, v_{i,2}, v_{i,3})$ in exponential coordinates and homogeneous 3D coordinates $\mathbf{X}_j = (X_j^{\text{world}}, Y_j^{\text{world}}, Z_j^{\text{world}}, W_j^{\text{world}})^\top$. The state vector has $3m$ elements from the rotations and $4n$ elements from the homogeneous coordinates. In each row of the Jacobian, the only non-zero elements are the ones that correspond to the rotation of the current camera i and the point j. That is, the Jacobian is a sparse matrix where every two rows are of the form

$$\begin{bmatrix} \cdots & \frac{\partial r_{ij,x}}{\partial v_{i,1}} & \frac{\partial r_{ij,x}}{\partial v_{i,2}} & \frac{\partial r_{ij,x}}{\partial v_{i,3}} & \cdots & \frac{\partial r_{ij,x}}{\partial X_j^{\text{world}}} & \frac{\partial r_{ij,x}}{\partial Y_j^{\text{world}}} & \frac{\partial r_{ij,x}}{\partial Z_j^{\text{world}}} & \frac{\partial r_{ij,x}}{\partial W_j^{\text{world}}} & \cdots \\ \cdots & \frac{\partial r_{ij,y}}{\partial v_{i,1}} & \frac{\partial r_{ij,y}}{\partial v_{i,2}} & \frac{\partial r_{ij,y}}{\partial v_{i,3}} & \cdots & \frac{\partial r_{ij,y}}{\partial X_j^{\text{world}}} & \frac{\partial r_{ij,y}}{\partial Y_j^{\text{world}}} & \frac{\partial r_{ij,y}}{\partial Z_j^{\text{world}}} & \frac{\partial r_{ij,y}}{\partial W_j^{\text{world}}} & \cdots \end{bmatrix}. \tag{3.12}$$

Deriving an expression for the partial derivative for any of the state variables q, we get

$$\frac{\partial r_{ij,x}}{\partial q} = \frac{\partial}{\partial q}\left(x_{\text{match}} - \left(f_x(x_{\text{sphere}}\frac{\rho}{r}) + c_x\right)\right) = -f_x\frac{\partial}{\partial q}\left(\rho\frac{x_{\text{sphere}}}{r}\right), \tag{3.13}$$

and applying the multivariable chain rule yields

$$\frac{\partial r_{ij,x}}{\partial q} = -f_x \left(\rho \frac{\partial}{\partial q} \left(\frac{x_{\text{sphere}}}{r} \right) + \frac{x_{\text{sphere}}}{r} \frac{\partial \rho}{\partial q} \right). \tag{3.14}$$

For the first summand we can rewrite

$$\frac{x_{\text{sphere}}}{r} = \frac{X}{\sqrt{X^2 + Y^2}} \tag{3.15}$$

and then express the derivative as

$$\frac{\partial}{\partial q} \left(\frac{X}{\sqrt{X^2 + Y^2}} \right) = \frac{\partial}{\partial X} \left(\frac{X}{\sqrt{X^2 + Y^2}} \right) \frac{\partial X}{\partial q} + \frac{\partial}{\partial Y} \left(\frac{X}{\sqrt{X^2 + Y^2}} \right) \frac{\partial Y}{\partial q}. \tag{3.16}$$

The derivatives of X, Y and Z with respect to q will be derived later in the text. For now we can evaluate

$$\begin{cases} \dfrac{\partial}{\partial X} \left(\dfrac{X}{\sqrt{X^2 + Y^2}} \right) = \dfrac{1}{\sqrt{X^2 + Y^2}} - X \dfrac{1}{2} \dfrac{2X}{(X^2 + Y^2)^{3/2}} = \dfrac{Y^2}{(X^2 + Y^2)^{3/2}} \\ \dfrac{\partial}{\partial Y} \left(\dfrac{X}{\sqrt{X^2 + Y^2}} \right) = -\dfrac{XY}{(X^2 + Y^2)^{3/2}} \end{cases}. \tag{3.17}$$

We can get the derivatives of $r_{ij,y}$ instead by simply switching X for Y and vice versa in the above expressions. For the second term in (3.14), the derivative of ρ can be expanded as

$$\frac{\partial \rho}{\partial q} = \frac{\partial \rho}{\partial \theta} \frac{\partial \theta}{\partial q}, \tag{3.18}$$

where

$$\frac{\partial \rho}{\partial \theta} = 1 + 3k_1\theta^2 + 5k_2\theta^4 + 7k_3\theta^6 + 9k_4\theta^8 \tag{3.19}$$

and

$$\frac{\partial \theta}{\partial q} = \frac{\partial \theta}{\partial X} \frac{\partial X}{\partial q} + \frac{\partial \theta}{\partial Y} \frac{\partial Y}{\partial q} + \frac{\partial \theta}{\partial Z} \frac{\partial Z}{\partial q}. \tag{3.20}$$

Using the expression for θ from (3.11) we get

$$\begin{cases} \dfrac{\partial \theta}{\partial X} = \dfrac{1}{\left(\frac{X}{Z}\right)^2 + \left(\frac{Y}{Z}\right)^2 + 1} \cdot \frac{1}{2} \dfrac{1}{\sqrt{\left(\frac{X}{Z}\right)^2 + \left(\frac{Y}{Z}\right)^2}} \cdot \dfrac{2X}{Z^2} = \dfrac{1}{X^2 + Y^2 + Z^2} \dfrac{XZ}{\sqrt{X^2 + Y^2}} \\ \dfrac{\partial \theta}{\partial Y} = \dfrac{1}{X^2 + Y^2 + Z^2} \dfrac{YZ}{\sqrt{X^2 + Y^2}} \\ \dfrac{\partial \theta}{\partial Z} = \dfrac{Z^2}{X^2 + Y^2 + Z^2} \cdot \frac{1}{2} \dfrac{Z}{\sqrt{X^2 + Y^2}} \cdot (-2) \dfrac{X^2 + Y^2}{Z^3} = -\dfrac{\sqrt{X^2 + Y^2}}{X^2 + Y^2 + Z^2} \end{cases}. \tag{3.21}$$

Now the only unknown expressions are $\frac{\partial X}{\partial q}$, $\frac{\partial Y}{\partial q}$ and $\frac{\partial Z}{\partial q}$. For $q \in \mathbf{v}$, these are elements of the matrix

$$\frac{\partial(P\mathbf{X})}{\partial \mathbf{v}} = \frac{\partial(R(\mathbf{v})\mathbf{X}^{X,Y,Z} + \mathbf{t}W)}{\partial \mathbf{v}} = \begin{bmatrix} \frac{\partial X}{\partial v_1} & \frac{\partial X}{\partial v_2} & \frac{\partial X}{\partial v_3} \\ \frac{\partial Y}{\partial v_1} & \frac{\partial Y}{\partial v_2} & \frac{\partial Y}{\partial v_3} \\ \frac{\partial Z}{\partial v_1} & \frac{\partial Z}{\partial v_2} & \frac{\partial Z}{\partial v_3} \end{bmatrix}. \tag{3.22}$$

The camera centers $\mathbf{C}$ are constants, but $t = -R\mathbf{C}$ is dependent of R. Therefore we have

$$\frac{\partial(P\mathbf{X})}{\partial \mathbf{v}} = \frac{\partial R(\mathbf{v})(\mathbf{X}^{X,Y,Z} - W\mathbf{C})}{\partial \mathbf{v}}, \tag{3.23}$$

which using the result from [11] can be rewritten as

$$\frac{\partial R(\mathbf{v})(\mathbf{X}^{X,Y,Z} - W\mathbf{C})}{\partial \mathbf{v}} = -R\left[(\mathbf{X}^{X,Y,Z} - W\mathbf{C})\right]_\times \frac{\mathbf{v}\mathbf{v}^\top + (R^\top - I)\left[\mathbf{v}\right]_\times}{||\mathbf{v}||^2}. \tag{3.24}$$

For $q \in \mathbf{X}$, we get

$$\frac{\partial(P\mathbf{X})}{\partial \mathbf{X}} = P = \begin{bmatrix} \frac{\partial X}{\partial X^{\text{world}}} & \frac{\partial X}{\partial Y^{\text{world}}} & \frac{\partial X}{\partial Z^{\text{world}}} & \frac{\partial X}{\partial W^{\text{world}}} \\ \frac{\partial Y}{\partial X^{\text{world}}} & \frac{\partial Y}{\partial Y^{\text{world}}} & \frac{\partial Y}{\partial Z^{\text{world}}} & \frac{\partial Y}{\partial W^{\text{world}}} \\ \frac{\partial Z}{\partial X^{\text{world}}} & \frac{\partial Z}{\partial Y^{\text{world}}} & \frac{\partial Z}{\partial Z^{\text{world}}} & \frac{\partial Z}{\partial W^{\text{world}}} \end{bmatrix}. \tag{3.25}$$

Now we have all of the expressions needed to calculate the Jacobian.

3.5.3 Implementation

The state vector to be optimized consists of the 3D points and the camera rotations rewritten in exponential coordinates (initialized from the CAD model values). When converting the initial rotation matrices into exponential coordinates using (2.15) we require a nonzero θ from (2.14). If $\theta = 0$ we add a small perturbation to the rotation. Note also that we optimize over the homogeneous coordinates, i.e. a four number representation (X, Y, Z, W) of the 3D points. This is for numerical reasons. Points may end up being far away during the optimization, and we would like to avoid having large values in the state vector. Homogeneous coordinates allow to preserve the numerical scale while the cartesian coordinates $(X/W, Y/W, Z/W)$ change size.

One might ask whether it is necessary to also optimize the 3D points, as opposed to only optimizing for the rotations. The alternative would be to, in every iteration of the optimization, triangulate 3D points from the matches. This is undesirable, as some points may end up behind the cameras and cause weird reprojections, impeding the optimization process. Also, the analytical Jacobian calculation above would no longer hold, since $\mathbf{X}$ would not be independent of the rotations. Therefore it is deemed more beneficial to optimize both rotations and 3D points.

The trust region reflective algorithm is used to optimize the least squares problem. In each step, the residual and Jacobian matrices are calculated using the expressions derived in the previous sections, with one exception. We optimize over the

normalized coordinates, i.e. pixel coordinates rescaled by the focal length and principal point. This is partly for numerical reasons, but also because we want to weigh the errors equally for all cameras independent of their resolutions. This way we essentially optimize the camera angles rather than the pixel values.

The optimization is divided into several stages, as listed in Table 3.2. The used loss functions are linear ($f(z) = z$), and Tukey [18], defined as

$$f(z) = \begin{cases} \frac{C^2}{6}\left(1 - \left(1 - \frac{z^2}{C^2}\right)^3\right) & \text{if } |z| < C, \\ \frac{C^2}{6} & \text{otherwise,} \end{cases} \tag{3.26}$$

with $C = 4.685$ as suggested in [19] (for data of unit variance). The sum of squares we had in (3.2) is the optimization problem with linear loss function. For any loss function f we instead solve

$$\min_{\mathbf{q}} \sum_k f(r_k(\mathbf{q})^2) \tag{3.27}$$

for elements r_k in the residual vector. The function scale s listed in Table 3.2 is used to rescale the loss function, which is supposed to be used for data of variance 1, so that we instead use

$$\overline{f}(z) = s^2 f\left(\frac{z^2}{s^2}\right). \tag{3.28}$$

This essentially defines what we consider to be an acceptable reprojection error for the current stage. For this reason all points that yield a higher reprojection error after the optimization stage are also rejected. The specific values in Table 3.2 are received through experimentation, from which these values seemed to give satisfactory results. Since the focal length of the fisheye cameras is roughly 400 pxl/m the values x from $s = \frac{x}{400}$ can be interpreted as the pixel equivalent in the fisheye cameras. After some of the iterations we also filter out points that are behind any of the cameras or further away than 20 m.

The motivation for this optimization method is that we in general will have bad matches among the point correspondences. Therefore we apply a robust loss function, in this case Tukey, at different function scales in order to reduce the residual lengths for the good matches, and discard the bad matches. The Tukey loss function at our different scales is shown in Figure 3.4. The idea is that the optimization will decrease the lengths of residuals in the steep (high derivative) areas for the good correspondences, and leave the bad ones behind to be discarded. After that the linear steps are supposed to weigh the remaining points equally.

Table 3.2: Optimization stages with a consecutive narrowing of the allowed residual lengths. After each stage we filter away matches that give too large reprojection errors. The loss functions are rescaled by the function scale. Note that this has no effect on linear loss. Filter by distance after indicates whether or not to remove points behind cameras or further away than 20 m.

Stage 1 Loss: Tukey Function scale: $\frac{10}{400}$ Filter by distance after: No
Stage 2 Loss: Tukey Function scale: $\frac{5}{400}$ Filter by distance after: No
Stage 3 Loss: Tukey Function scale: $\frac{2}{400}$ Filter by distance after: Yes
Stage 4 Loss: Linear Function scale: $\frac{2}{400}$ Filter by distance after: Yes
Stage 5 Loss: Linear Function scale: $\frac{2}{400}$ Filter by distance after: No

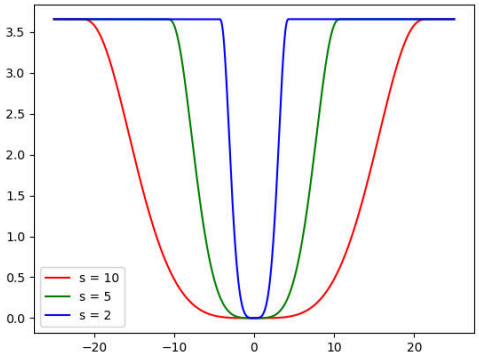


Figure 3.4: The Tukey loss function with function scale s set to 10, 5 and 2 pixels. The horizontal axis shows residual lengths, and the loss function is applied to the squared residuals. Note that the loss function has derivative zero at the tails and is very flat close to the center.

3.6 Evaluation

For a qualitative overview of the accuracy of the method, birds eye view images are generated using the fisheye camera poses from the CAD model, our optimization method and other available calibrations (factory calibration, dynamic pattern rig or the motion estimation method in [1]). This is done using image sets with a relatively flat ground containing lane markings, shadows or other features that indicate whether the overlapping parts are well aligned.

In order to create the birds eye view images, the fisheye images are first undistorted and projected to the flat ground plane, which is assumed to be parallel to the horizontal part of the car coordinate system, and offset vertically by a constant height. The images are projected to a square of size 20 m by 20 m centered at the car coordinate system. Note that in doing so, we also apply an interpolation to replace missing pixels in the projected image. Then the pixels outside of a 180° field of view are turned black to simplify interpreting the images. We combine the projected images from all four cameras by simply averaging over the pixel values. The resulting birds eye view image thus has four brighter areas with the pairwise overlapping parts of the camera views. An example is shown in Figure 3.5.

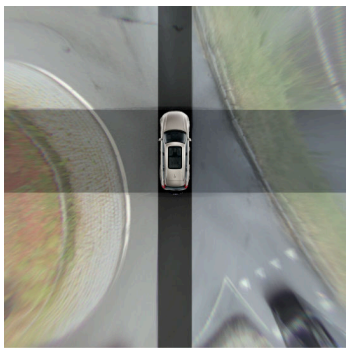
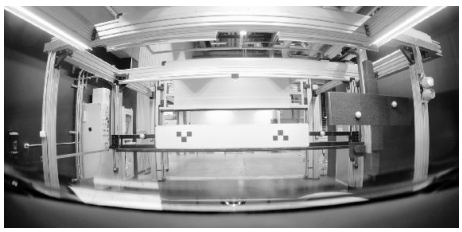


Figure 3.5: Example of a birds eye view image. An image of a car body is included in the middle to specify the orientation. The darker rectangles are areas that only use projections from a single camera, and the brighter parts contain overlapping camera views.

This is a fast and simple way to compare calibration accuracies. However, there are some important drawbacks from using this method. It requires a good estimate of the height of the vehicle coordinate system above the ground. This is not a fix value, but might vary with e.g. the suspension system, tire pressure and cargo weight. It also assumes that the car body is parallel to the ground, which might not be true if the car carries an uneven load distribution. Even if it was, the ground is often not perfectly flat. We assume a flat ground plane, and can thus only expect points that actually lie in this plane to align for calibrated cameras. Objects above the ground plane, on the other hand, will project in different directions for different cameras. For an intuitive understanding, you can think of the projection on the ground plane as the shadow the objects would cast on the ground if the camera was a light source.

In order to get a more quantitative evaluation metric, some point correspondences are annotated manually. In the evaluation step, they are triangulated and then re-projected into the camera view. This is done for the CAD model poses, the currently used calibration poses and our optimized poses. This reprojection error is used as an accuracy metric. Note that the accuracy of these annotations is also limited, probably to around $0.5 - 1.0$ pixels. For each scenario, 10 such correspondences are annotated for each camera pair.

The extrinsics calibrations currently used on the cars come from one out of two sources. For some scenarios the poses are from a factory calibration, i.e. a static, pattern based calibration that is performed once in the factory when a new car is built. For the other scenarios the poses are from a dynamic pattern based calibration station, where the car is standing on a rotating platform with lots of different target patterns around it. This type of calibration is performed regularly. The calibration stations are shown in Figure 3.6. For some scenarios we can also access results from the aforementioned motion based method in [1].



(a) Factory calibration station (forward view).

(b) The dynamic pattern-based calibration station.

Figure 3.6: The pattern-based calibration stations. Their output calibrations are used for comparing the accuracy to our method.

4

Results

The results consist of birds eye view images and reprojection errors from manually annotated point correspondences. For each scenario, we use 10 annotated correspondences for each camera pair. All annotation reprojection errors are given in units of fisheye image pixels. For the long range cameras their errors are rescaled to the equivalent number of fisheye camera pixels (without the radial distortion) for interpretability. Then we average over all camera pairs. This is calculated using both the CAD model poses, the current calibration used in the car and our optimized calibration.

The data is split into some sets with only fisheye images and some with all cameras included, mainly because of the uncertainty that follows from non consistent timestamps with the long range cameras. Also, the surround view system in itself is a recurrent research subject, which allows for comparison between this work and some of the research described in section 2.6. Within these sections we also split the data into some sets where the factory calibration is given, and some where the latest dynamic pattern calibration is given, in order to compare performance to both types.

4.1 Surround View System

For the following, only fisheye camera images are used in the scenarios.

4.1.1 Comparing with Factory Calibration

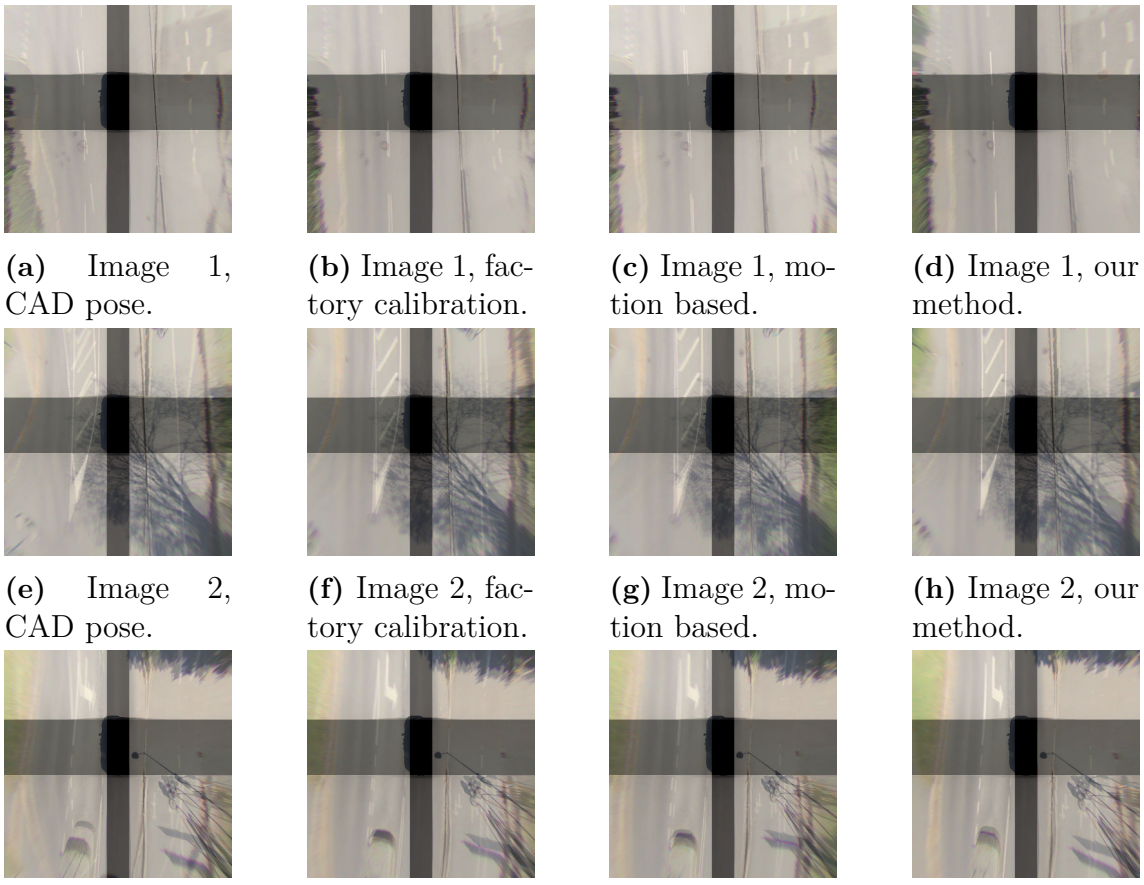
The reprojection errors are listed in Table 4.1, along with links to the corresponding birds eye view images. Table 4.2 shows the average angle error between the factory calibration and our calibration. The relative angles for all cameras are listed in Appendix 2.

Table 4.1: Average reprojection error (and standard deviation in parentheses) over the annotated correspondences using camera poses from the CAD model, the factory calibration, the motion based method in [1] and our optimization method. Only fisheye cameras are included in this part.

Scenario index	CAD model poses	Factory calibration	Motion estimation	Our calibration	Birds eye view
1	4.20 (1.53)	2.24 (2.35)	1.65 (1.10)	0.66 (0.62)	Figure 4.1
2	3.06 (2.36)	1.49 (1.03)	2.49 (1.15)	0.67 (0.46)	Figure 4.2
3	2.35 (2.82)	1.53 (1.05)	3.72 (1.56)	1.02 (0.73)	Figure 4.3
4	7.84 (4.20)	4.28 (3.62)	1.87 (1.25)	0.91 (0.70)	Figure 4.4
5	6.01 (4.88)	3.17 (2.56)	2.11 (1.17)	0.97 (0.69)	Figure 4.5
6	4.03 (2.43)	1.28 (0.83)	1.85 (1.48)	0.85 (0.66)	Figure 4.6
7	4.32 (1.88)	2.40 (1.40)	1.76 (0.88)	0.88 (0.63)	Figure 4.7
8	4.77 (3.18)	2.67 (2.05)	1.87 (1.30)	0.86 (0.71)	Figure 4.8
9	4.77 (2.14)	2.26 (1.71)	1.37 (1.06)	0.70 (0.52)	Figure 4.9
10	6.04 (4.49)	4.15 (2.55)	1.18 (0.76)	0.83 (0.79)	Figure 4.10
Overall	**4.74**	**2.55**	**1.99**	**0.84**	-

Table 4.2: Average absolute value of the relative rotation angles (in degrees) between the factory calibration and our calibration for scenario 1-10 with only fisheye cameras.

Scenario index	Angle in x-direction	Angle in y-direction	Angle in z-direction
1	0.9	0.3	0.26
2	0.51	0.28	0.93
3	0.54	0.29	0.62
4	0.75	0.33	0.69
5	0.56	0.26	0.52
6	0.24	0.18	0.72
7	0.39	0.36	0.22
8	0.3	0.34	0.48
9	0.63	0.3	0.41
10	0.59	0.36	0.29

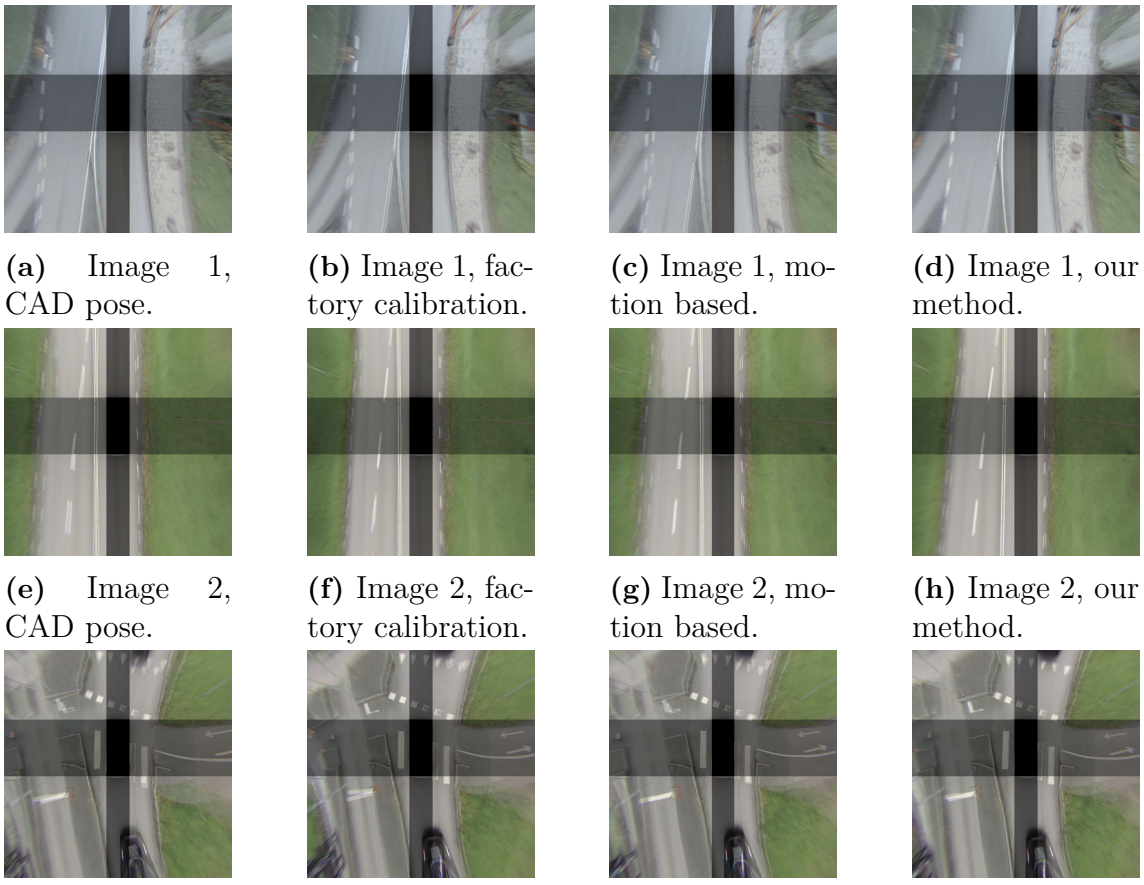


(a) Image 1, CAD pose. **(b)** Image 1, factory calibration. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, factory calibration. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, factory calibration. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.1: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 1 with only fisheye images. Note that there is an elevated sidewalk in the right part of each image, so we can not expect lines to overlap there. Lines on the road should align.

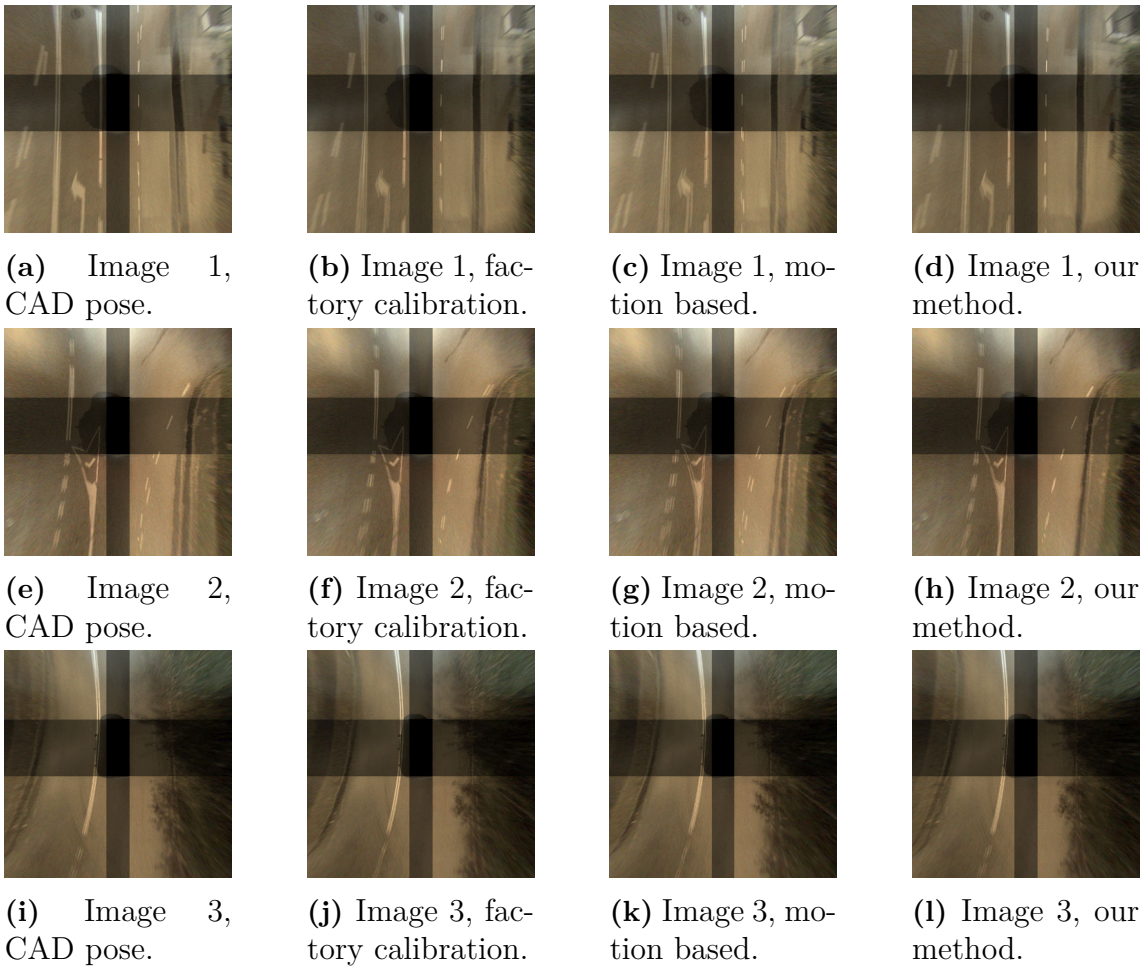


(a) Image 1, CAD pose. **(b)** Image 1, factory calibration. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, factory calibration. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, factory calibration. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.2: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 2 with only fisheye images.

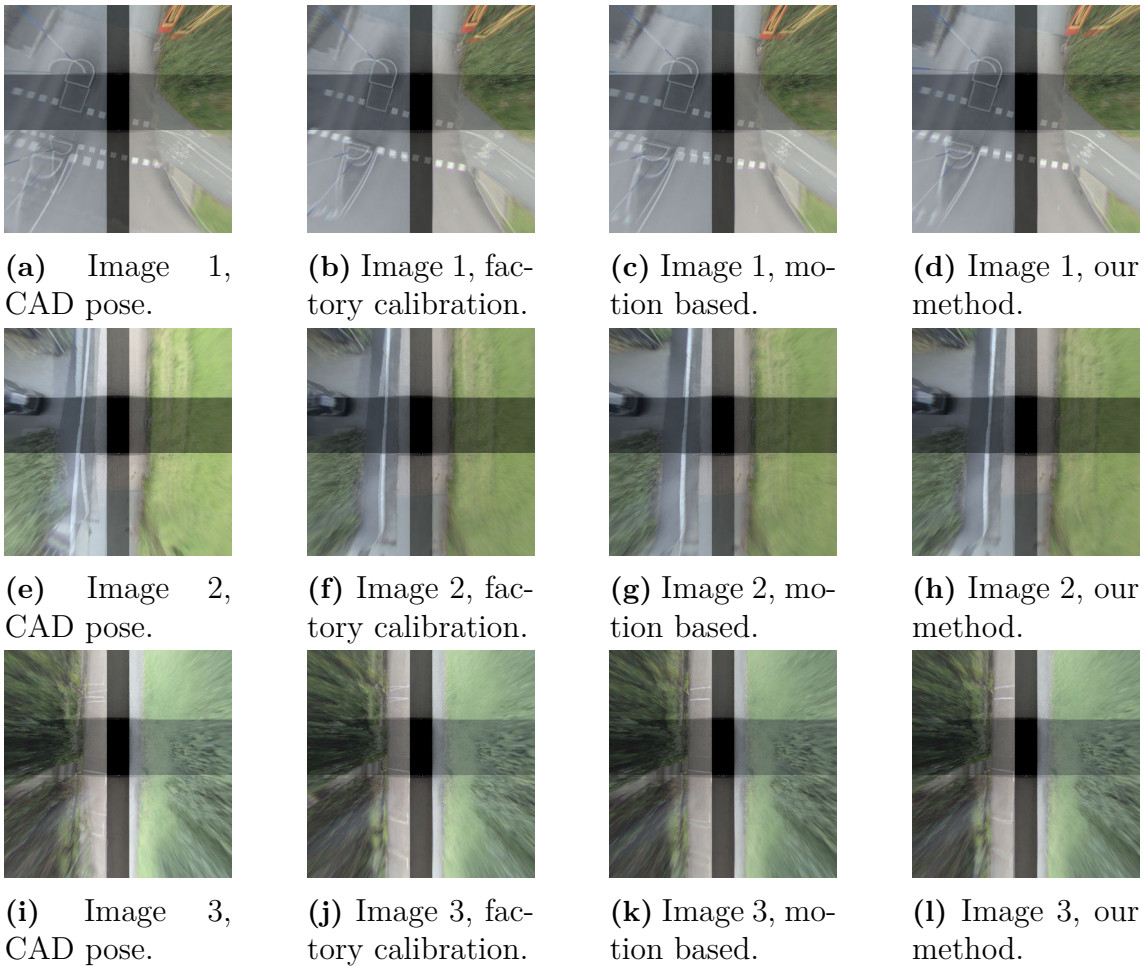


(a) Image 1, CAD pose. **(b)** Image 1, factory calibration. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, factory calibration. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, factory calibration. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.3: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 3 with only fisheye images.

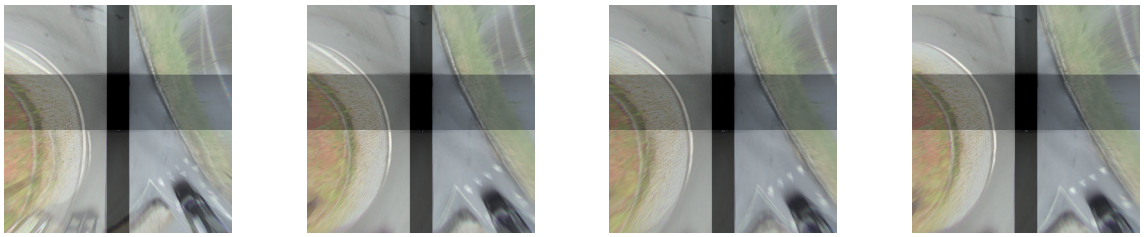


(a) Image 1, CAD pose. **(b)** Image 1, factory calibration. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, factory calibration. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, factory calibration. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.4: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 4 with only fisheye images.

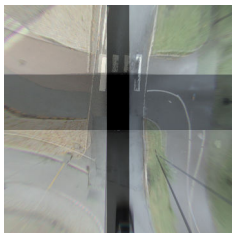


(a) Image 1, CAD pose.

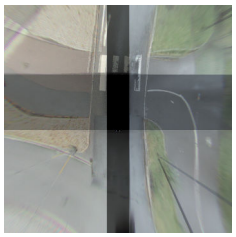
(b) Image 1, factory calibration.

(c) Image 1, motion based.

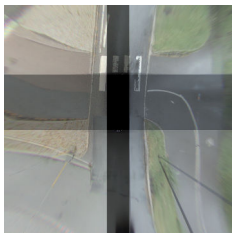
(d) Image 1, our method.



(e) Image 2, CAD pose.

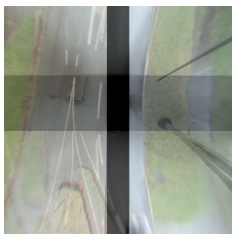


(f) Image 2, factory calibration.

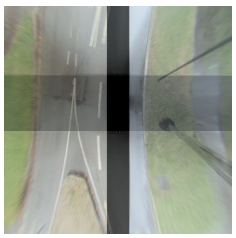


(g) Image 2, motion based.

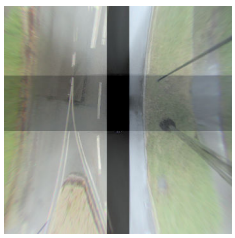
(h) Image 2, our method.



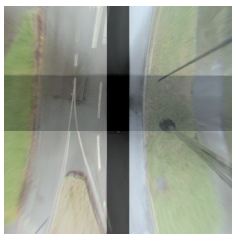
(i) Image 3, CAD pose.



(j) Image 3, factory calibration.



(k) Image 3, motion based.



(l) Image 3, our method.

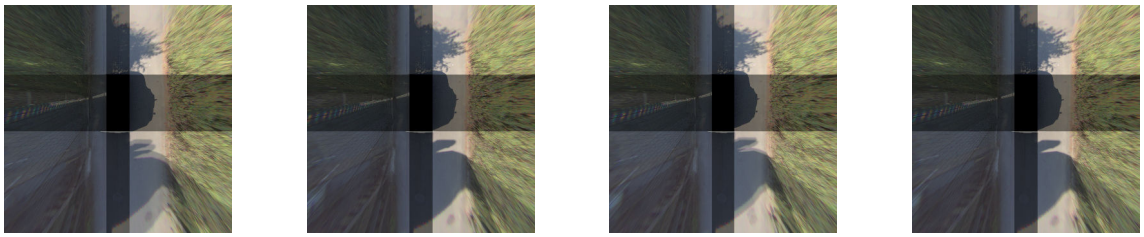
Figure 4.5: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 5 with only fisheye images.

(a) Image 1, CAD pose.

(b) Image 1, factory calibration.

(c) Image 1, motion based.

(d) Image 1, our method.

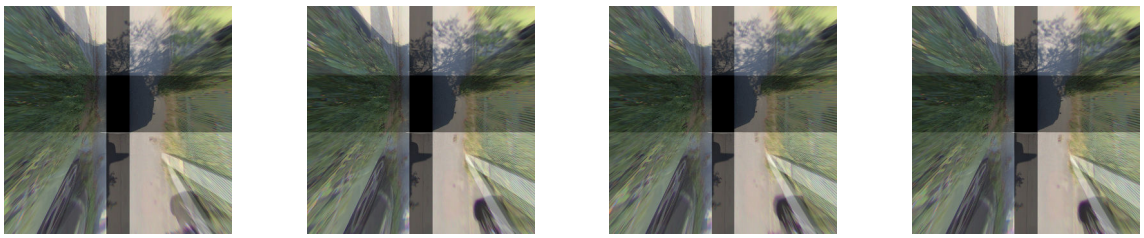


(e) Image 2, CAD pose.

(f) Image 2, factory calibration.

(g) Image 2, motion based.

(h) Image 2, our method.



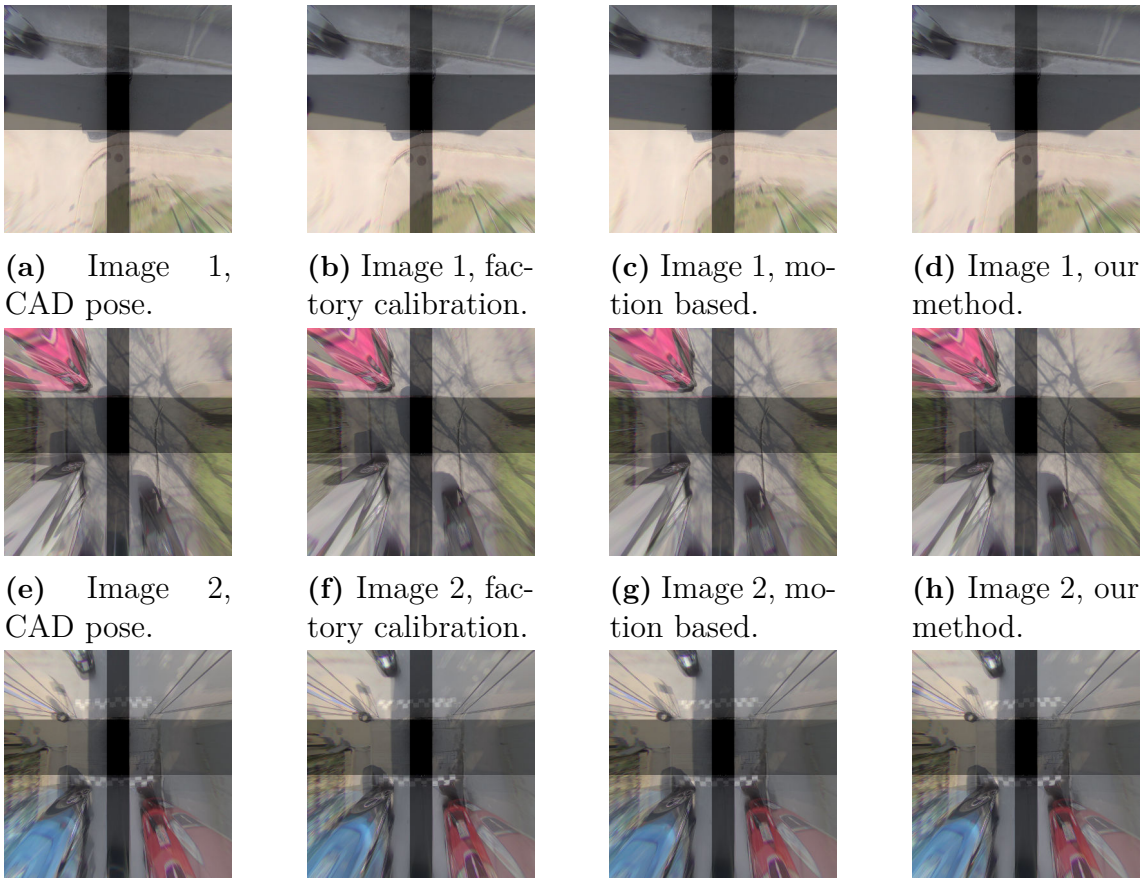
(i) Image 3, CAD pose.

(j) Image 3, factory calibration.

(k) Image 3, motion based.

(l) Image 3, our method.

Figure 4.6: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 6 with only fisheye images.

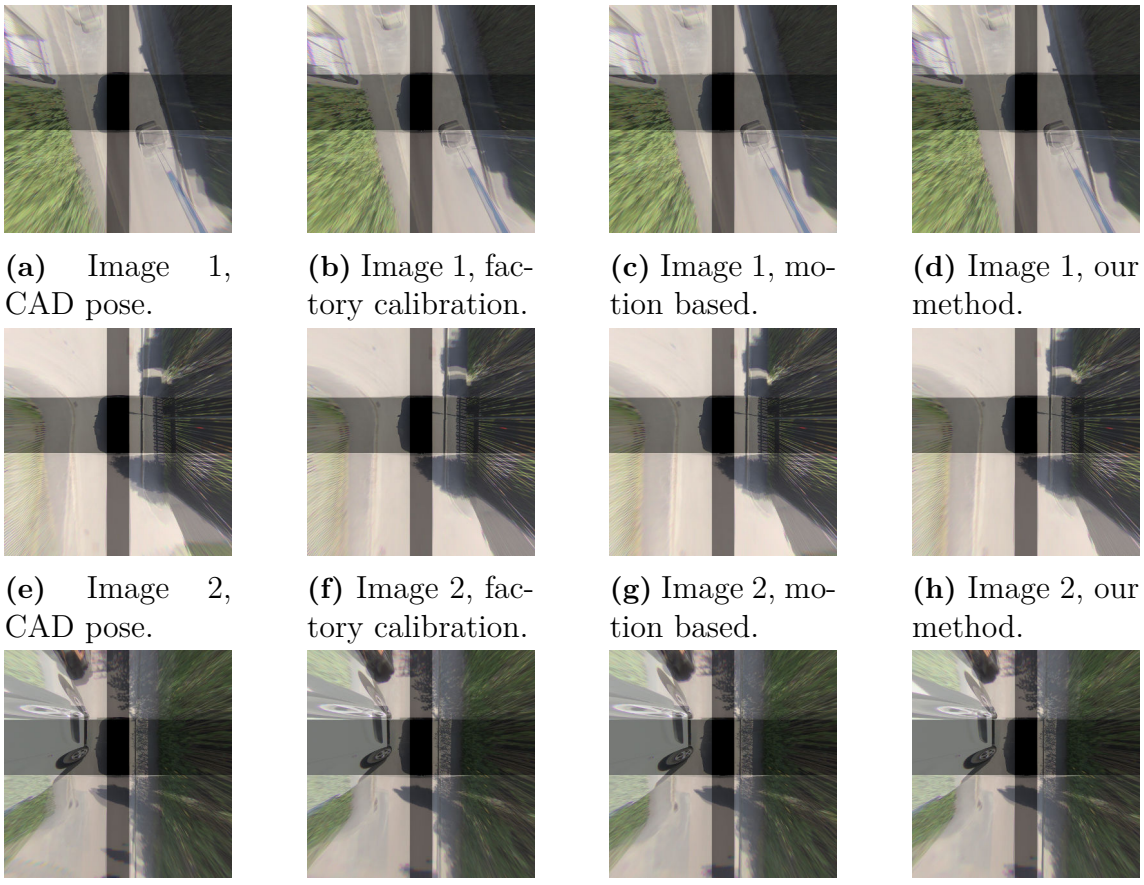


(a) Image 1, CAD pose. **(b)** Image 1, factory calibration. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, factory calibration. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, factory calibration. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.7: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 7 with only fisheye images.

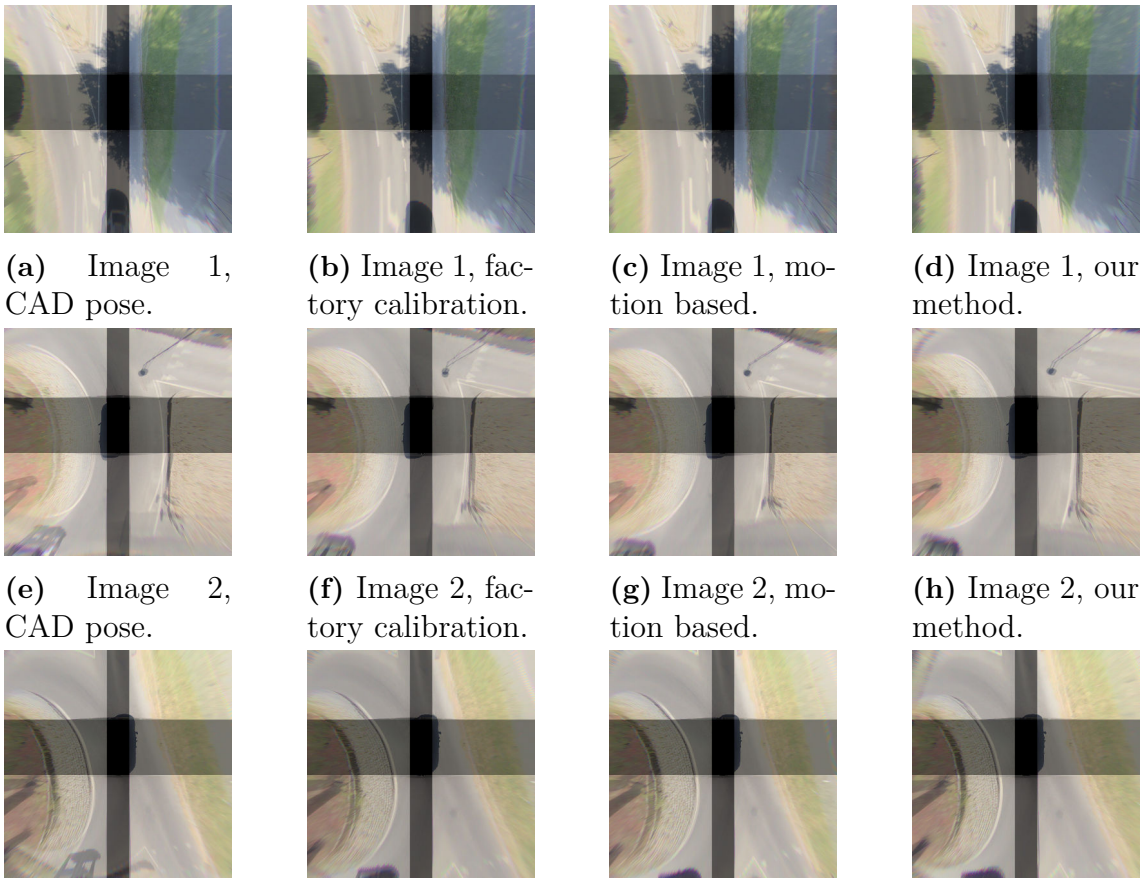


(a) Image 1, CAD pose. **(b)** Image 1, factory calibration. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, factory calibration. **(g)** Image 2, motion based. **(h)** Image 2, our method.

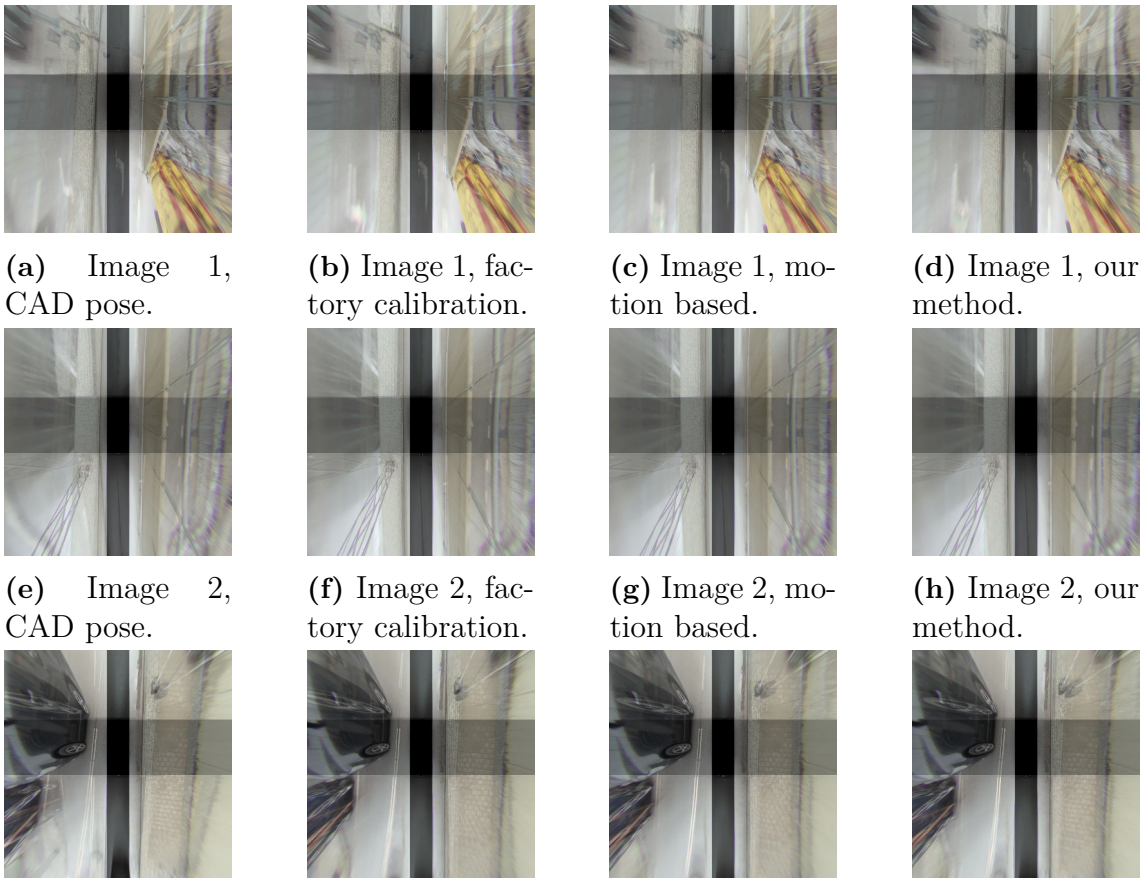
(i) Image 3, CAD pose. **(j)** Image 3, factory calibration. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.8: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 8 with only fisheye images.



(a) Image 1, CAD pose.
(b) Image 1, factory calibration.
(c) Image 1, motion based.
(d) Image 1, our method.
(e) Image 2, CAD pose.
(f) Image 2, factory calibration.
(g) Image 2, motion based.
(h) Image 2, our method.
(i) Image 3, CAD pose.
(j) Image 3, factory calibration.
(k) Image 3, motion based.
(l) Image 3, our method.

Figure 4.9: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 9 with only fisheye images.



(a) Image 1, CAD pose. **(b)** Image 1, factory calibration. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, factory calibration. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, factory calibration. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.10: Birds eye view images created using the poses from the CAD model, the factory calibration, the motion based online calibration method in [1] and our optimization method for scenario 10 with only fisheye images.

4.1.2 Comparing with Dynamic Pattern Calibration

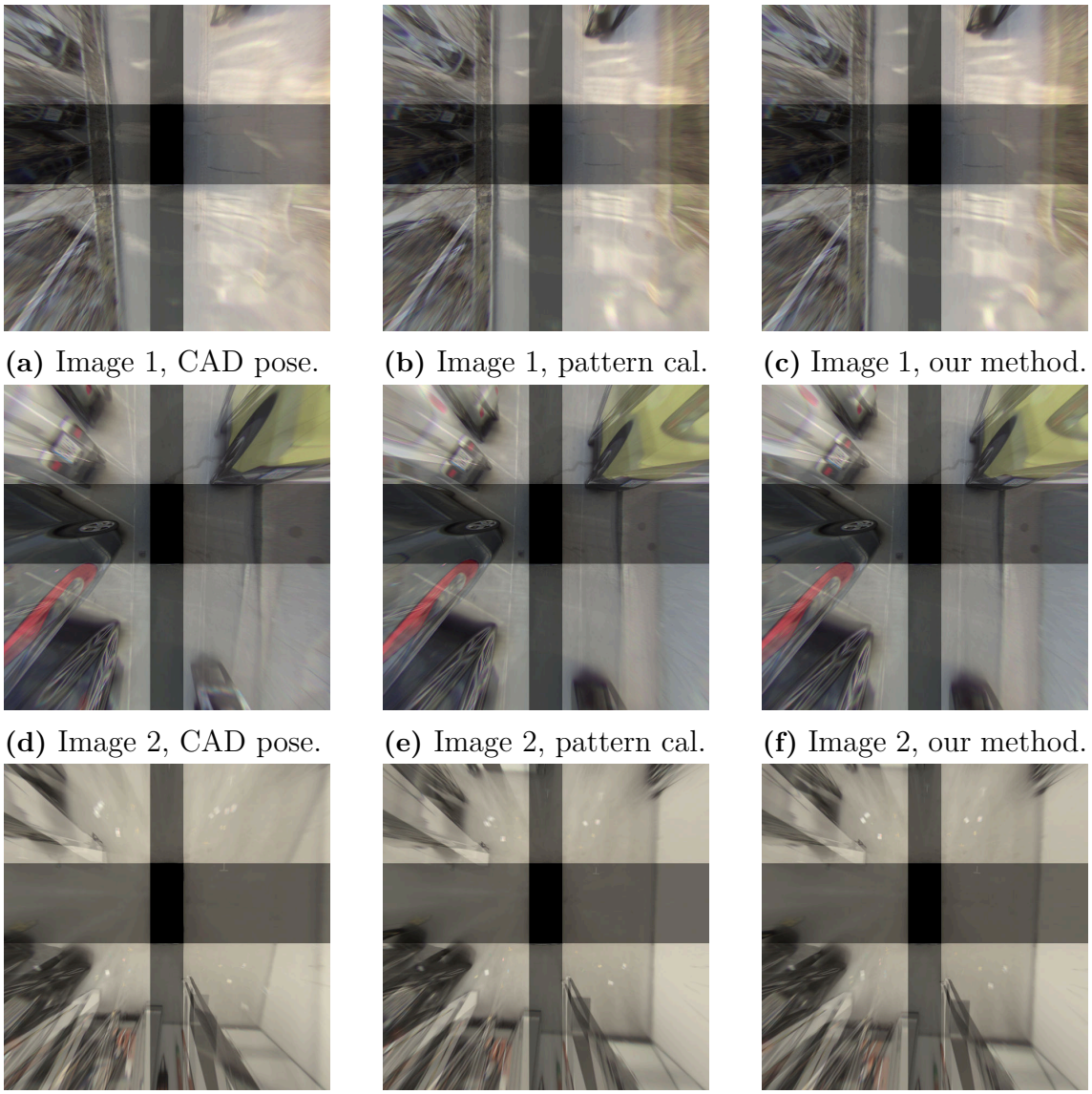
The reprojection errors are listed in Table 4.3, along with links to the corresponding birds eye view images. For scenarios 11, 12 and 14 the motion estimation results are not available due to missing data. Table 4.4 shows the average angle error between the dynamic pattern calibration and our calibration. The relative angles for all cameras are listed in Appendix 2.

Table 4.3: Average reprojection error (and standard deviation in parentheses) over the annotated correspondences using camera poses from the CAD model, the latest dynamic pattern calibration, the motion based method in [1] and our optimization method. Only fisheye cameras are included in this part.

Scenario index	CAD model poses	Pattern calibration	Motion estimation	Our calibration	Birds eye view
11	4.82 (4.89)	0.82 (0.58)	-	0.66 (0.59)	Figure 4.11
12	9.03 (5.56)	1.44 (1.06)	-	1.00 (0.75)	Figure 4.12
13	10.12 (2.95)	0.81 (0.53)	2.37 (1.48)	0.86 (0.61)	Figure 4.13
14	6.02 (3.84)	0.90 (0.59)	-	0.66 (0.52)	Figure 4.14
15	6.79 (4.01)	0.47 (0.32)	8.89 (8.62)	0.59 (0.41)	Figure 4.15
16	7.38 (2.77)	0.65 (0.78)	2.84 (1.65)	0.58 (0.91)	Figure 4.16
17	10.04 (3.26)	0.95 (0.69)	3.49 (2.43)	0.41 (0.32)	Figure 4.17
18	9.12 (6.91)	0.78 (0.56)	8.01 (5.90)	0.92 (0.82)	Figure 4.18
19	11.84 (9.23)	0.78 (0.55)	6.91 (5.65)	0.59 (0.58)	Figure 4.19
20	16.32 (9.09)	1.09 (0.71)	1.29 (0.65)	0.72 (0.61)	Figure 4.20
Overall	**9.15**	**0.87**	**4.83**	**0.71**	-

Table 4.4: Average absolute value of the relative rotation angles (in degrees) between the dynamic pattern calibration and our calibration for scenario 11-20 with only fisheye cameras.

Scenario index	Angle in x-direction	Angle in y-direction	Angle in z-direction
11	0.07	0.13	0.61
12	0.19	0.12	0.3
13	0.21	0.19	0.55
14	0.27	0.36	0.42
15	0.33	0.21	0.41
16	0.13	0.19	0.63
17	0.27	0.18	0.29
18	0.48	0.14	0.68
19	0.4	0.15	0.79
20	0.32	0.32	1.51

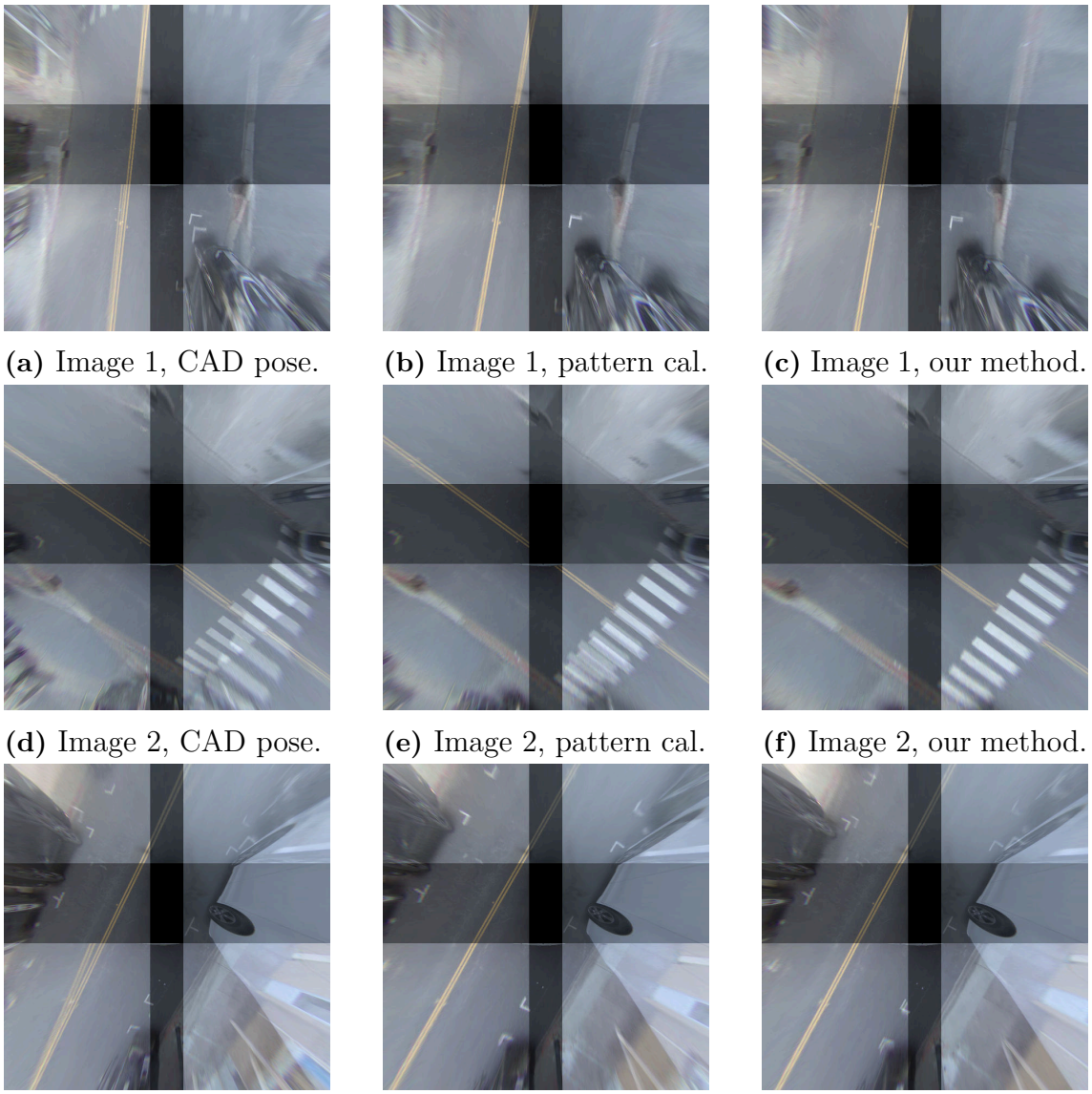


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure 4.11: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 11 with only fisheye images.

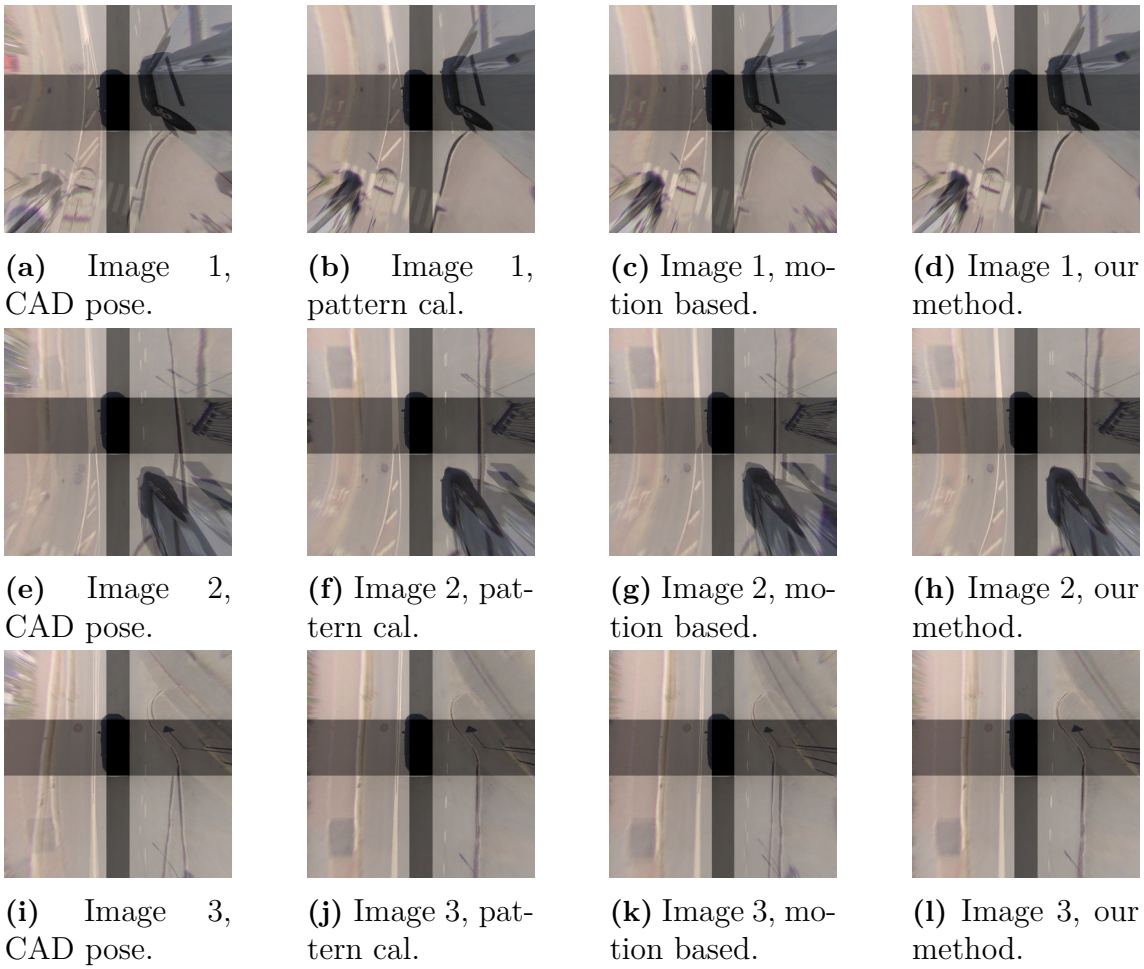


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure 4.12: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 12 with only fisheye images.

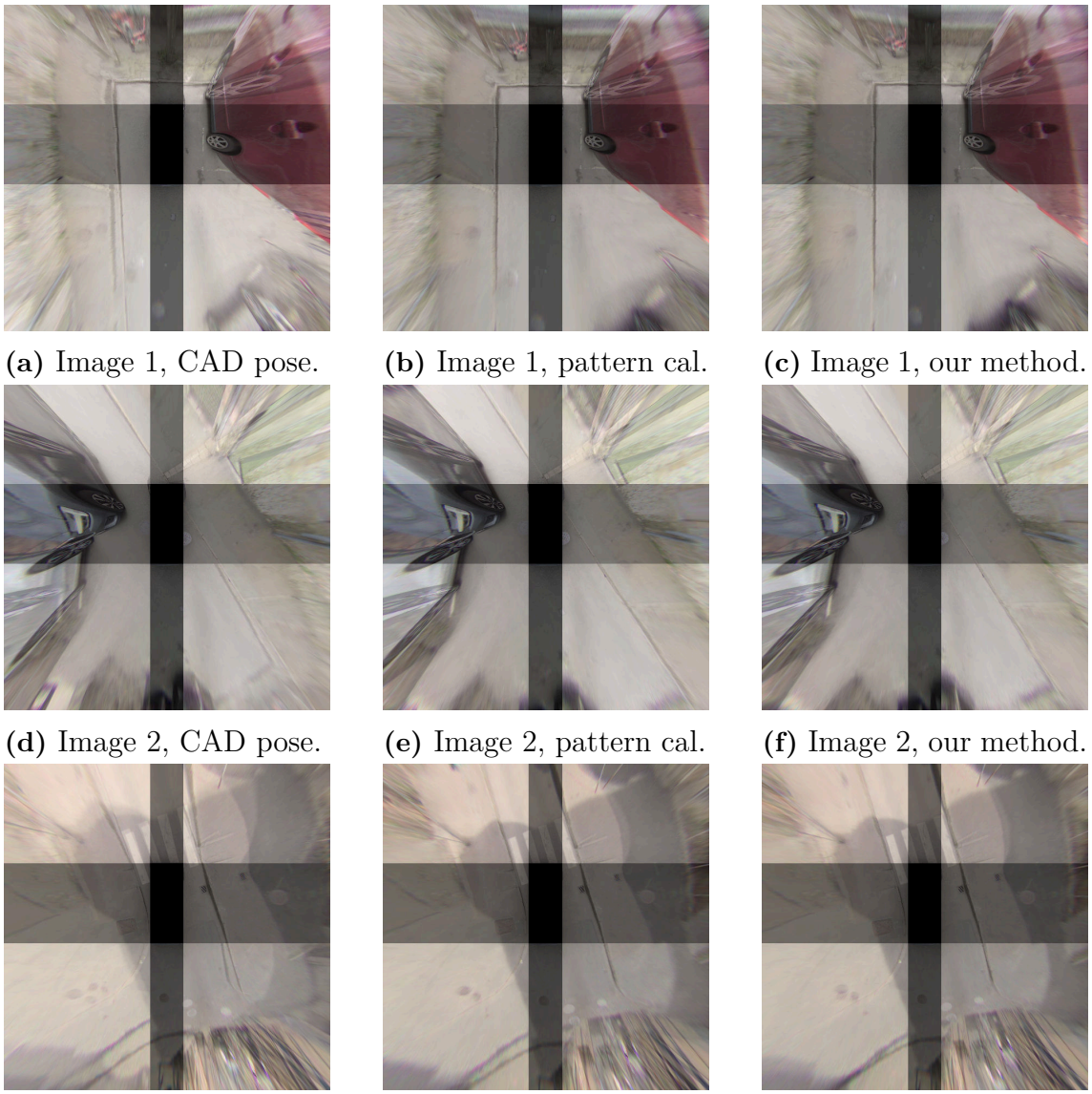


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, pattern cal. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, pattern cal. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.13: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration, the motion based online calibration method in [1] and our optimization method for dataset 13 with only fisheye images.

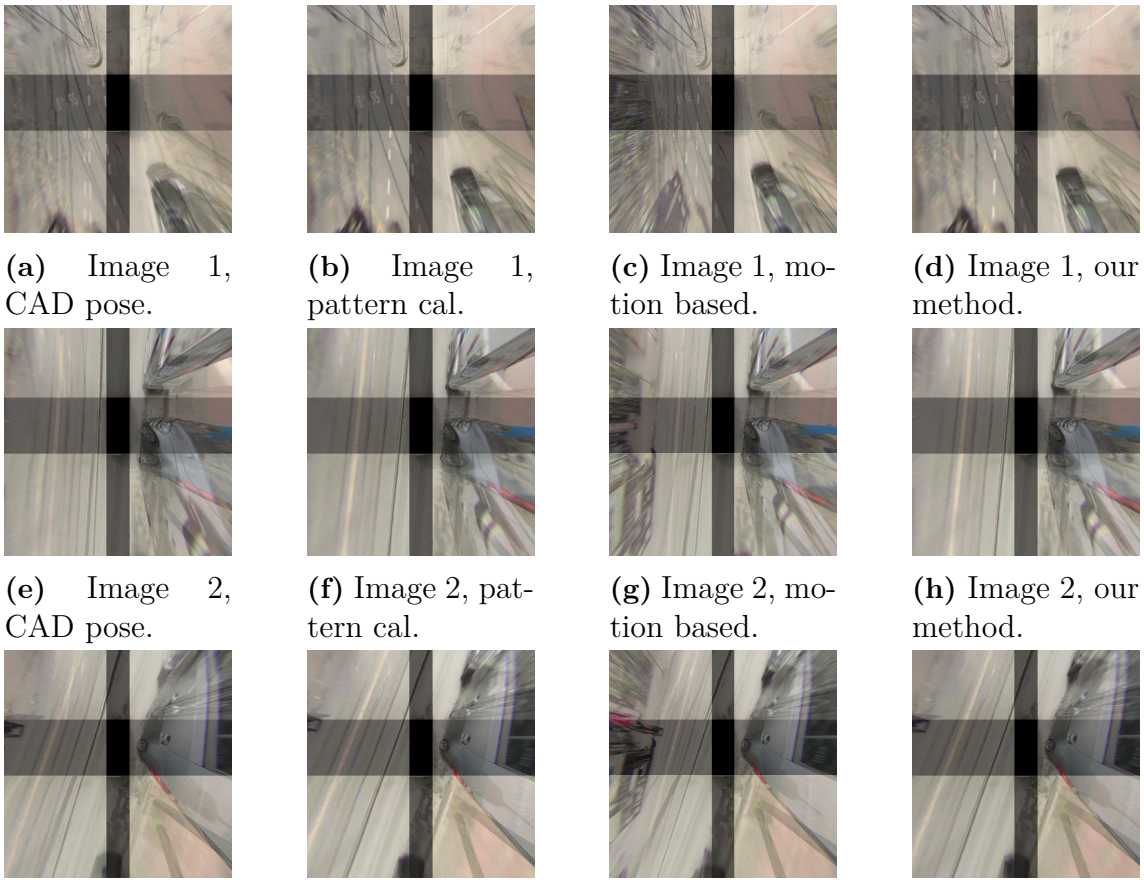


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure 4.14: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 14 with only fisheye images.

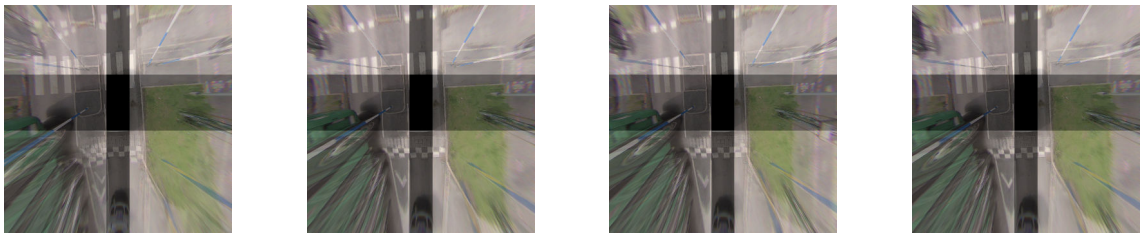


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, pattern cal. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, pattern cal. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.15: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration, the motion based online calibration method in [1] and our optimization method for dataset 15 with only fisheye images.

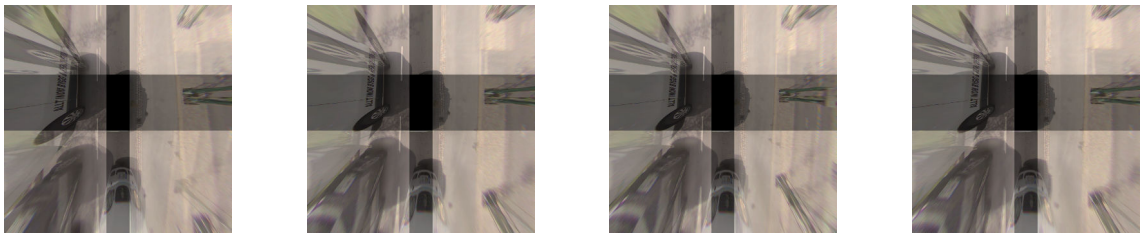


(a) Image 1, CAD pose.

(b) Image 1, pattern cal.

(c) Image 1, motion based.

(d) Image 1, our method.

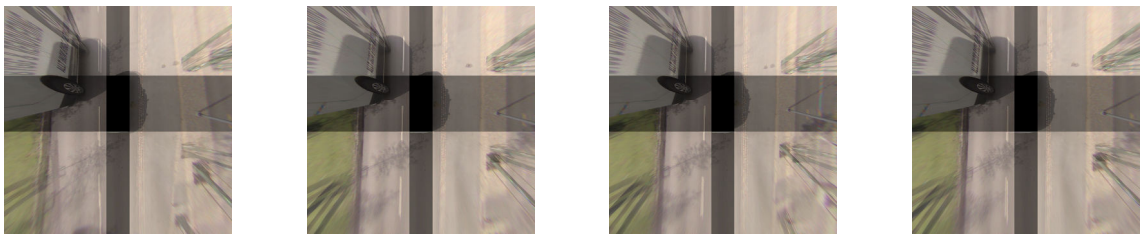


(e) Image 2, CAD pose.

(f) Image 2, pattern cal.

(g) Image 2, motion based.

(h) Image 2, our method.



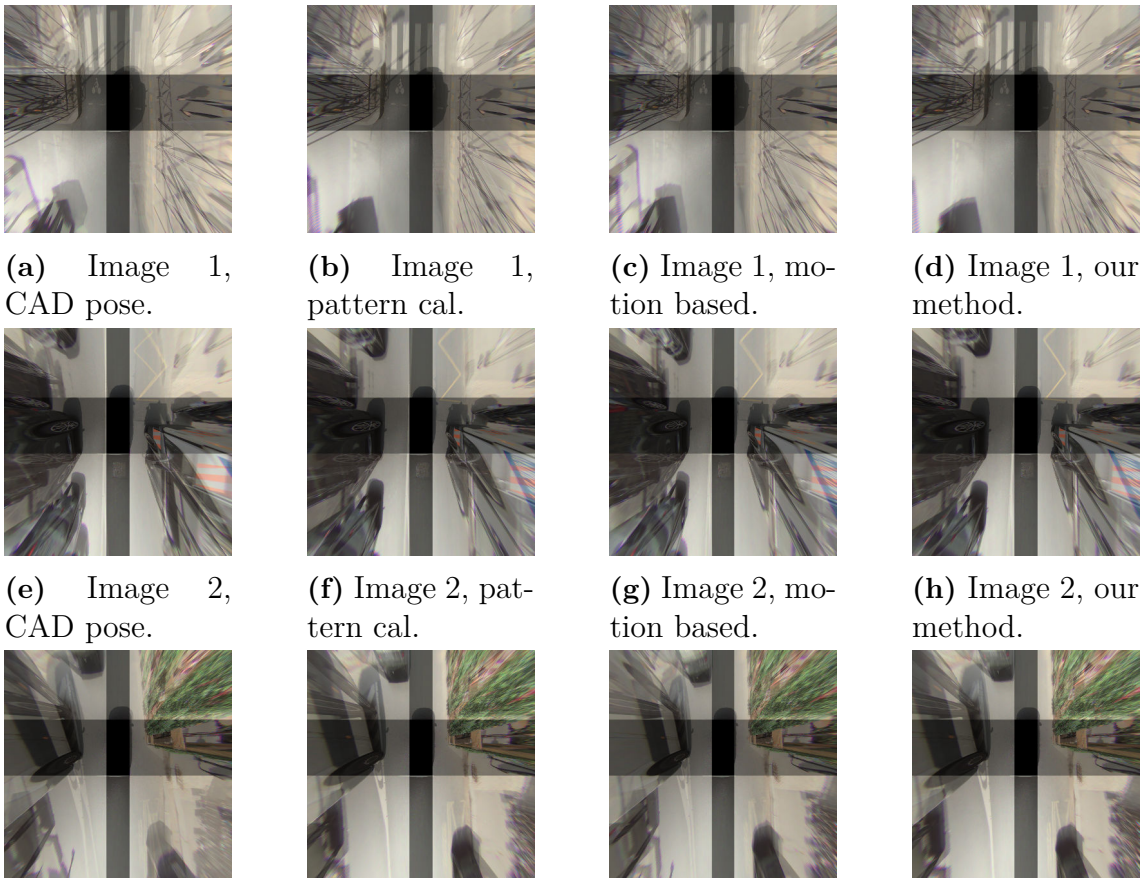
(i) Image 3, CAD pose.

(j) Image 3, pattern cal.

(k) Image 3, motion based.

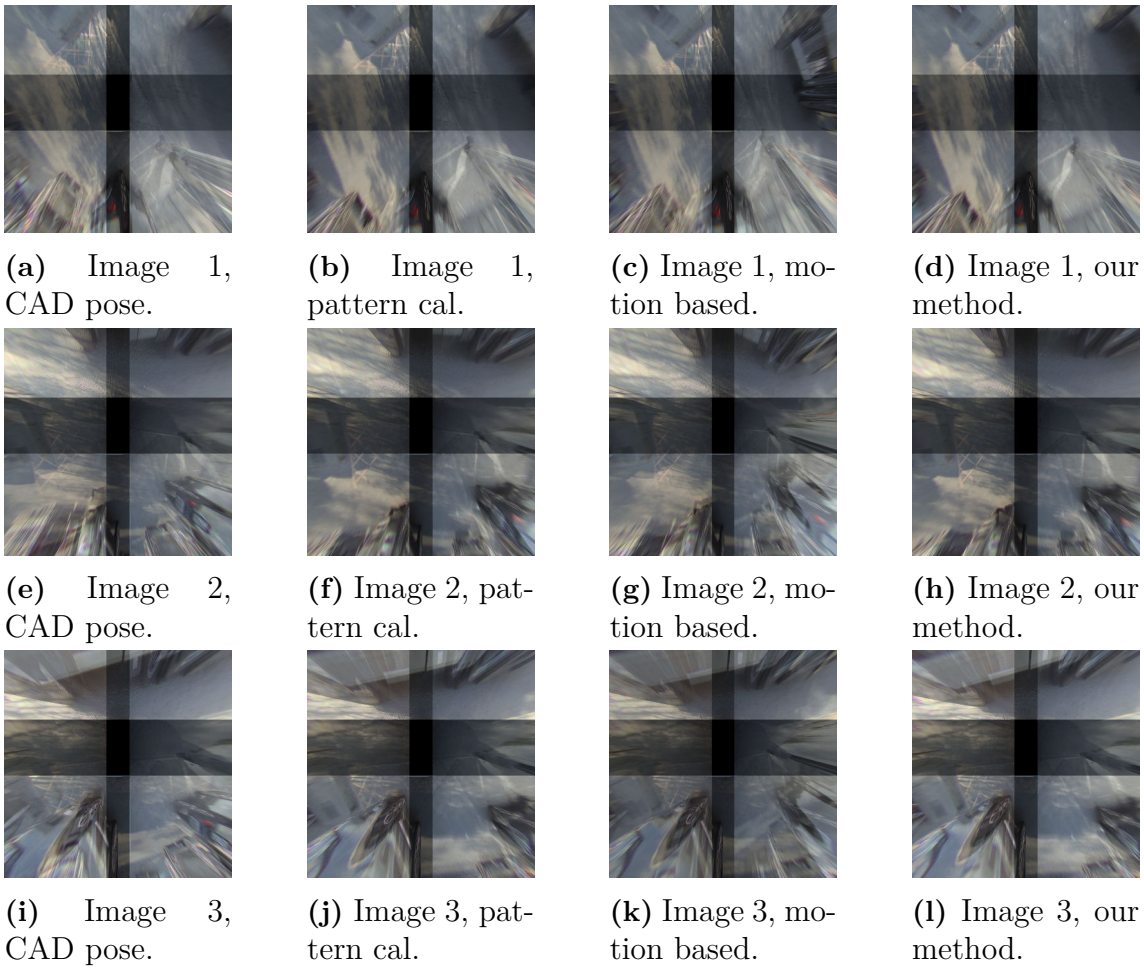
(l) Image 3, our method.

Figure 4.16: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration, the motion based online calibration method in [1] and our optimization method for dataset 16 with only fisheye images.



(a) Image 1, CAD pose.
(b) Image 1, pattern cal.
(c) Image 1, motion based.
(d) Image 1, our method.
(e) Image 2, CAD pose.
(f) Image 2, pattern cal.
(g) Image 2, motion based.
(h) Image 2, our method.
(i) Image 3, CAD pose.
(j) Image 3, pattern cal.
(k) Image 3, motion based.
(l) Image 3, our method.

Figure 4.17: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration, the motion based online calibration method in [1] and our optimization method for dataset 17 with only fisheye images.

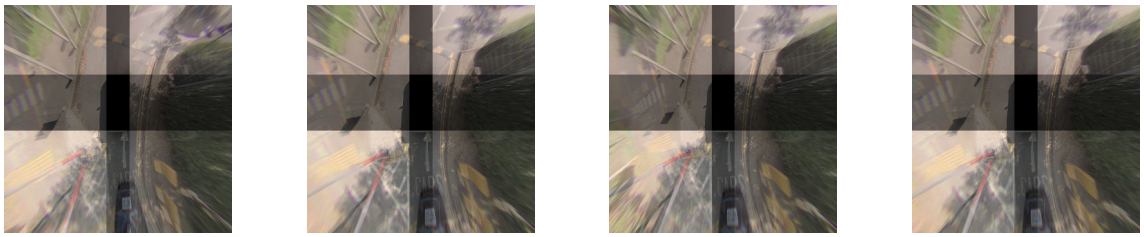


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, motion based. **(d)** Image 1, our method.

(e) Image 2, CAD pose. **(f)** Image 2, pattern cal. **(g)** Image 2, motion based. **(h)** Image 2, our method.

(i) Image 3, CAD pose. **(j)** Image 3, pattern cal. **(k)** Image 3, motion based. **(l)** Image 3, our method.

Figure 4.18: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration, the motion based online calibration method in [1] and our optimization method for dataset 18 with only fisheye images.

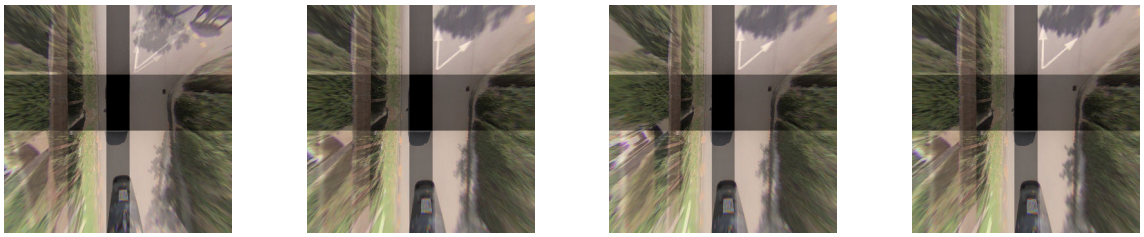


(a) Image 1, CAD pose.

(b) Image 1, pattern cal.

(c) Image 1, motion based.

(d) Image 1, our method.

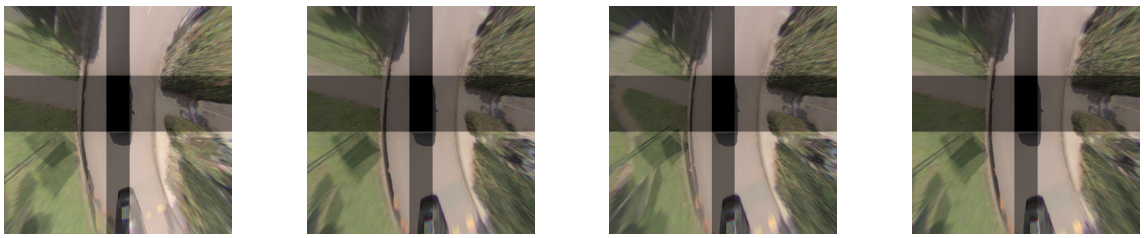


(e) Image 2, CAD pose.

(f) Image 2, pattern cal.

(g) Image 2, motion based.

(h) Image 2, our method.



(i) Image 3, CAD pose.

(j) Image 3, pattern cal.

(k) Image 3, motion based.

(l) Image 3, our method.

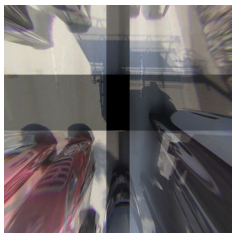
Figure 4.19: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration, the motion based online calibration method in [1] and our optimization method for dataset 19 with only fisheye images.

(a) Image 1, CAD pose.

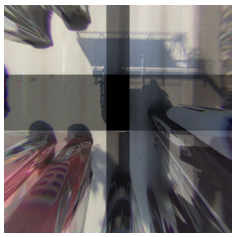
(b) Image 1, pattern cal.

(c) Image 1, motion based.

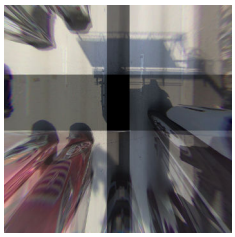
(d) Image 1, our method.



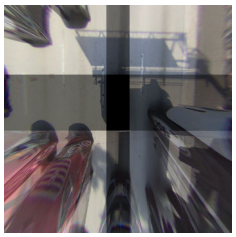
(e) Image 2, CAD pose.



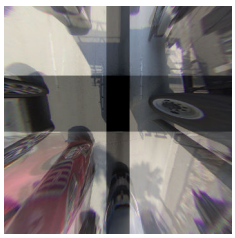
(f) Image 2, pattern cal.



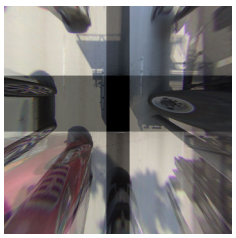
(g) Image 2, motion based.



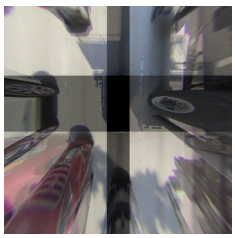
(h) Image 2, our method.



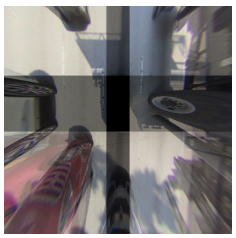
(i) Image 3, CAD pose.



(j) Image 3, pattern cal.



(k) Image 3, motion based.



(l) Image 3, our method.

Figure 4.20: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration, the motion based online calibration method in [1] and our optimization method for dataset 20 with only fisheye images.

4.2 All Cameras

The following scenarios contain images from all cameras. The birds eye view images are placed in Appendix 1.

4.2.1 Comparing with Factory Calibration

The reprojection errors are listed in Table 4.5, along with links to the corresponding birds eye view images. Table 4.6 shows the average angle error between the factory calibration and our calibration. The relative angles for all cameras are listed in Appendix 2.

Table 4.5: Average reprojection error (and standard deviation in parentheses) over the annotated correspondences using camera poses from the CAD model, the factory calibration and our optimization method. All cameras are included in the optimization and evaluation.

Scenario index	CAD model poses	Factory calibration	Our method	Birds eye view
21	4.27 (3.08)	1.00 (0.76)	0.83 (0.58)	Figure A.1
22	3.22 (2.64)	1.37 (2.18)	1.04 (1.98)	Figure A.2
23	3.45 (2.58)	1.12 (0.69)	0.33 (0.32)	Figure A.3
24	3.64 (2.15)	1.01 (0.73)	0.66 (0.59)	Figure A.4
25	3.42 (2.59)	1.13 (0.70)	0.61 (0.61)	Figure A.5
26	3.48 (2.58)	1.11 (0.67)	0.94 (0.55)	Figure A.6
27	3.49 (2.33)	0.62 (0.67)	0.83 (0.72)	Figure A.7
28	3.46 (2.33)	0.62 (0.70)	0.72 (0.65)	Figure A.8
29	5.88 (2.13)	0.72 (0.49)	0.52 (0.54)	Figure A.9
30	3.43 (2.43)	0.70 (1.01)	0.46 (0.91)	Figure A.10
Overall	**5.59**	**0.95**	**0.76**	-

Table 4.6: Average absolute value of the relative rotation angles (in degrees) between the factory calibration and our calibration for scenario 21-30 with all cameras included.

Scenario index	Angle in x-direction	Angle in y-direction	Angle in z-direction
21	0.27	0.16	0.15
22	0.25	0.15	0.50
23	0.34	0.27	0.35
24	0.17	0.17	0.30
25	0.21	0.24	0.30
26	0.11	0.30	0.24
27	0.21	0.19	0.34
28	0.36	0.12	1.15
29	0.17	0.20	0.28
30	0.34	0.20	0.51

4.2.2 Comparing with Dynamic Pattern Calibration

The reprojection errors are listed in Table 4.7, along with links to the corresponding birds eye view images. Table 4.8 shows the average angle error between the factory calibration and our calibration. The relative angles for all cameras are listed in Appendix 2.

Table 4.7: Average reprojection error (and standard deviation in parentheses) over the annotated correspondences using camera poses from the CAD model, the latest dynamic pattern calibration and our optimization method. All cameras are included in the optimization and evaluation.

Scenario index	CAD model poses	Pattern calibration	Our method	Birds eye view
31	5.60 (4.46)	0.59 (0.49)	0.60 (0.51)	Figure A.11
32	7.74 (5.59)	1.23 (1.01)	1.48 (1.22)	Figure A.12
33	8.79 (6.21)	1.15 (1.02)	0.99 (0.96)	Figure A.13
34	8.38 (3.62)	0.88 (0.66)	0.43 (0.42)	Figure A.14
35	6.50 (5.79)	0.68 (0.56)	0.61 (0.57)	Figure A.15
36	6.18 (4.45)	0.66 (0.59)	0.68 (0.65)	Figure A.16
37	3.74 (2.10)	0.80 (0.75)	0.91 (0.83)	Figure A.17
38	6.35 (3.10)	0.69 (0.67)	0.56 (0.72)	Figure A.18
39	10.72 (6.43)	0.97 (0.76)	0.96 (0.91)	Figure A.19
40	7.53 (5.33)	1.04 (0.99)	1.01 (0.97)	Figure A.20
Overall	**7.15**	**0.87**	**0.94**	-

Table 4.8: Average absolute value of the relative rotation angles (in degrees) between the dynamic pattern calibration and our calibration for scenario 31-40 with all cameras included.

Scenario index	Angle in x-direction	Angle in y-direction	Angle in z-direction
31	0.30	0.24	0.07
32	1.03	0.97	0.53
33	0.53	0.22	0.50
34	0.26	0.34	0.32
35	0.10	0.22	0.49
36	0.34	0.28	0.21
37	0.40	0.20	0.38
38	0.36	0.22	0.83
39	0.67	0.25	0.61
40	0.18	0.16	0.29

5

Discussion

Based on the results, we begin this chapter by again discussing the used evaluation metrics and how to interpret them. We argue that both the birds eye view images and annotated reprojection errors can be used to determine the accuracy of a calibration, and that using both of them is beneficial for the purpose of verifying that our method works. Additionally, we need the reprojection error for the scenarios using all cameras, since the long range cameras are not used in the birds eye view images. With this in mind, we then compare the accuracy of the different extrinsic calibrations within each scenario. It seems that we achieve a higher accuracy than the motion estimation method in [1], and similar accuracy to the dynamic pattern-based calibration rig. Then we compare our results to those of other articles that are especially focused towards calibrating a surround view system, i.e. only fish-eye cameras. Our method seems to achieve similar accuracy as other state of the art methods, while also being more generally applicable. Finally, we discuss the specifics of our method, including its parameter setup, computational efficiency and limitations.

5.1 Evaluation Metrics

From the results, there are a couple of things to notice. Firstly, a lower reprojection error on the annotated data seems to correlate with better looking birds eye view images, i.e. ones where lines on the flat road plane align in the overlapping parts of the images. Look for example at Figure 4.1, Figure 4.4 and Figure 4.5. The respective reprojection errors (in Table 4.1) are substantially lower for our optimized poses than for the factory calibration. At the same time the birds eye view images have distinctly better alignment. This confirms that both of these metrics to some degree can be used to evaluate the accuracy of a calibration.

Even though they are used for the same cause, it seems advantageous to include both metrics in the evaluation. The birds eye view images enable a qualitative assessment of the accuracy. Also, since the intended use case of the fisheye camera rig is mainly to create a surround view around the car, it makes sense to look directly at these images. However, as previously mentioned we assume a flat ground plane in the creation of the birds eye view images. This is often a good approximation in close proximity around the car, but further away it is typically less accurate. For example, many roads are sloped towards the middle lane marking (e.g. for drainage reasons), so that the other lane might be better approximated by a plane with another normal direction. This is also the case for many parking spots. Even a few centimeters of height difference may produce visible misalignment in the birds eye view images, so one needs to be quite careful when looking at these with the purpose of evaluating performance.

Another issue with birds eye view images is parts of the environment that are not even close to the ground plane, for example sidewalks and other cars. These are projected differently for each camera, which means the corresponding areas in the birds eye view images can not be used for evaluation, since they are not expected to overlap. This is a recurring problem since there are often sidewalks or elevated grass areas to the right side of the car, e.g. in Figure 4.1 and Figure 4.2. Thus there might be a bias towards evaluating the left side more often.

The annotation reprojection error, on the other hand, is a quantitative metric. To some extent it provides a direct way to see which calibration performs better for a certain scenario. However, we can in general not compare the values between different scenarios, since the annotation accuracy might vary. Also, as stated previously we can not expect the accuracy of the annotations to be lower than approximately $0.5 - 1.0$ pixels. Note also that we have quite high standard deviation in the annotated reprojection errors, which likely is an indication that some annotations are quite inaccurate. Thus, for some scenarios it might be hard to draw conclusions about which calibration is better, and one might need to consider them being equally good. Apart from that, the manual annotation is quite limiting for real use cases, but is useful for verifying that the method works.

Neither of the two metrics can be used to automatically tell the accuracy of a given

calibration, as both require human input. Ideally one should have such a method available if the method was to be implemented as part of the company software, in order to avoid using a bad calibration in the car. However, the method finds a better calibration compared to the initial guess (CAD model poses) in all the tested scenarios. The only time the results have been off is when some camera pair has (close to) 0 matches after the last optimization stage. Thus we have a criterion for when the optimization did not work as expected, and may simply put a threshold for the minimum number of matches left in each camera pair to approve the calibration. The problem can usually be solved by simply using more images for feature matching.

One could try other methods for evaluation, such as the aforementioned photometric error which is supposed to measure how well aligned a birds eye view image is. Thus it is somewhat limited by the flat ground plane approximation. Also it is sensitive to the illumination and exposure time of the different cameras [13], so it might require additional altering of the images or camera parameters. Another possible method is using RANSAC and epipolar geometry to find a set of inliers among some matched features, and then calculate their reprojection error, as is typical in many computer vision pipelines [2]. This is quite similar to our pipeline and also requires feature matching, so if there for some reason are many wrong matches offset in a certain direction the results might be misleading, but one might want to consider it for future work.

5.2 Accuracy of our Calibrations

Here we start by considering the results from the scenarios containing only fisheye camera images. The annotation reprojection errors in Table 4.1 show clear improvements in the accuracy using our method compared to the factory calibration. The optimized poses give substantially lower errors, which seem to be on the order of the expected annotation accuracy. The corresponding birds eye view images also indicate better accuracy for our optimized poses. One should probably expect the factory calibration to be quite offset in most cases, since the cars have driven for a substantial amount of time since it was carried out. As for the motion estimation method in [1], which calibrates each camera separately, we also seem to get consistently better accuracy both for the annotation reprojection error and the birds eye view images.

Comparing to the dynamic pattern calibration it is not as clear which calibration gives better accuracy. The annotation reprojection errors in Table 4.3 are similar for all scenarios. The birds eye view images also look very similar, albeit varying a bit in sharpness between the calibrations. Thus we seem to achieve an accuracy comparable to the dynamic pattern based calibration for these scenarios. Here we also get better accuracy than the motion estimation method in [1].

For the scenarios with all cameras included we have an extra uncertainty in the camera translations due to the inconsistent timestamps of images. Since the scenarios are taken where the car moves slowly the error should be relatively small, but in general not negligible. The translation might differ by a couple of centimeters in the driving direction. Comparing the reprojection errors to the factory calibration (Table 4.5) we seem to get a bit better accuracy using our method for most scenarios, and similar accuracy for one of them. The birds eye view images all look very similar. This is not surprising since all of this data uses a factory calibration performed only a few months prior to collecting the data.

The dynamic pattern calibration accuracy is unclear when comparing to all cameras as well. In some cases the pattern-based method works better and in some our method works better, as can be seen in Table 4.7. For the birds eye view images in Figure A.11 and Figure A.15 the pattern-based calibration is more accurate, but for Figure A.13 ours looks a little bit better. However, since the birds eye view images only contain information from the fisheye cameras they are less comprehensive for evaluation. Thus it seems like the dynamic pattern calibration and our method achieve similar accuracies.

5.3 Comparing Accuracy to Related Work

For some more context we can also compare the results to some of the papers mentioned in 2.6 (discussing only the surround view system). Some of these articles use other quantitative error metrics such as photometric error [12] or mean distance error [13]. There are however some that list the angle differences to a ground truth pose given by a pattern-based calibration. They get an average error of 0.31° in [17] (the lane markings based method) and 0.55° in [14] (the method that uses virtual camera views from overlapping fields of view).

In our case, it does not seem reasonable to consider any of the pattern-based calibrations as a ground truth, since they were all made some time ago when the data was gathered. Thus they in general do not seem to give perfect overlap in the birds eye view images and our method often results in as good or better annotation reprojection errors. However, there are some scenarios where the pattern-based calibration was performed somewhat recently. Scenarios 21-30 all use a factory calibration from within a few months. The relative angles in Table 4.6 are all (with the exception of the z-angle for scenario 28) relatively low, with an average of 0.28° over all angles in the table. Also, some recent dynamic pattern-based calibrations are found in scenario 12, 15, 17, 34, 36 and 40, with relative angles of around the same size. Therefore it seems that we achieve an accuracy better than [14] and comparable to [17].

One important note is that in [17] they use an image resolution of 1280×720 pixels. We have 1936×1220 pixels. Therefore it is possible that their method might also yield around pixel-level accuracy. Of course the real world angle error is more important, but if they would be able to achieve similar accuracies by simply increasing the camera resolution the method itself could be deemed equally effective in that regard.

On the other hand, producing well aligned birds eye view images also requires calibration accuracy on the scale of the camera resolution. We can directly compare the quality of these images to some of the other articles. In [17] the birds eye view images align quite well, but still have some clearly visible small misalignment for some of their data. Note that this is also true for their so called ground truth angles, further highlighting that such pattern-based approaches also give a limited accuracy. The photometric optimization approach in [12] and the manual clicking method in [13] both seem to give satisfactory alignment in the overlap. Our method thus seems to have an accuracy comparable to those two articles.

5.4 Method Specifics

The method developed in this project is generally applicable to a wide range of environments, and does not require any special targets or landmarks for the calibration. Neither does it build on the assumption of a flat ground plane. The only critical requirement is that we should be able to find feature matches within 20 m from the respective camera pairs. The method might work poorly if using only images from wide open spaces such as large, empty parking spots. Then you would simply need to collect images for a longer time or perform the calibration in a different area.

As stated before, the ability to find the rotation of a camera pose is unaffected by the distance to the points. However, it is the fix translations that enforce the rotations to be given in the vehicle coordinate system. The reprojection of points further away are less affected by the translations, and instead depend more on only the relative rotations of the cameras. Thus, a scenario with only features far away from the car could yield a solution where the rotations are all correct relative to one another but offset in some direction relative to the car. Figure 5.1 shows two different sets of poses that could give nearly equivalent solutions for far away points. This is not only hypothetical. Running the program without the distance threshold we for some scenarios get solutions where the poses are visibly offset. In these cases the birds eye view images are naturally also skewed. The distance threshold is not necessarily required as long as we get some matches from points close to the car, but it guarantees that we will not find such offset rotations.

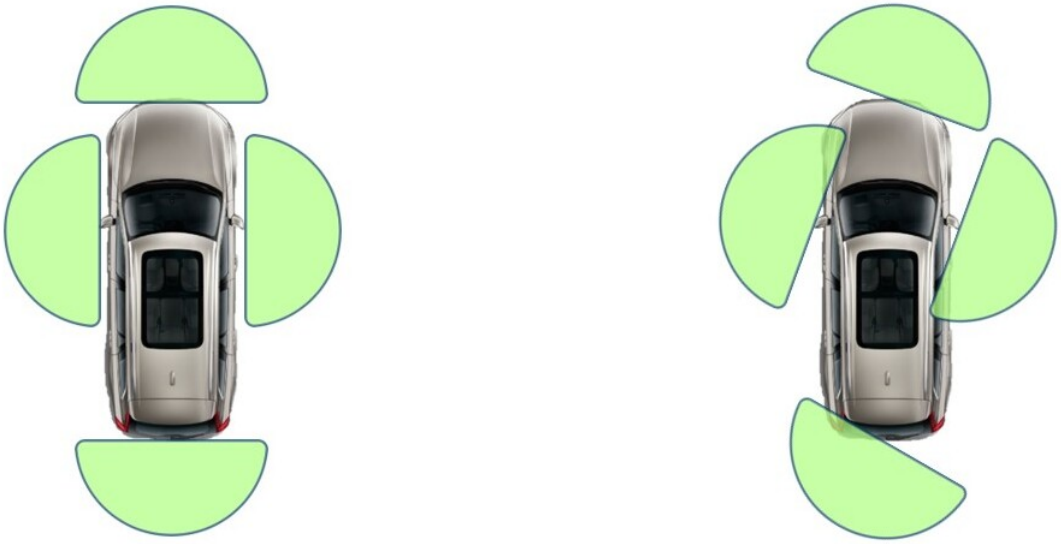


Figure 5.1: Example of two solutions that could give similar reprojection errors when the point correspondences are far away from the car. The relative rotations between cameras are the same for both cases, but the rotations are all offset clockwise in the right image.

As for the optimization, the results indicate that our successive lowering of function scale with Tukey loss function seems to work fine in rejecting bad correspondences. The exact settings can be discussed, however. The lower function scale is here set to 2 pixels. We want this value to be low enough so that bad matches are removed, but setting it too low might result in us choosing a subset of inliers where all points are shifted by noise in one direction. The chosen value seems to work, but in future work one might want to tune this along with some other parameters to increase performance. The function scale in the first optimization stage might need tuning as well. For example, if one were to find a situation where the CAD model pose is more offset from the true pose, then the residuals for the true matches could be outside of the loss function and all get rejected. Then there should be quite few correspondences left after the optimization. For future implementations, one possible solution when detecting such a case is to simply rerun the optimization with an added initial stage that uses Tukey loss function and a function scale of e.g. 15 pixels.

Computationally the method is not that demanding. The feature matching runs at over 30 frames per second for both the feature extractor [6] and the feature matcher [7], on their respective hardware setup. Most of the computation should be allocated in the optimization part. The program was implemented in python, using which the optimization takes around a minute, but it is hard to speculate about runtime if it were to be implemented as part of the company software. However, the current online calibration method takes a few minutes to run per camera, so even with our non-optimized, partly native python code we achieve quite a substantial speed up. The optimization part usually takes around 100-200 residual function evaluations in total. The total amount of computation is then of course dependent on the specific number of point correspondences in each optimization stage, but this should at least give a hint about the computational load.

Note also that the best achievable accuracy of our method is directly dependent on the accuracy of our feature extractor. Since we use ALIKED [6] we get sub-pixel coordinates for the feature correspondences. This is a clear advantage compared to getting discrete pixel coordinates both because of an increase in possible accuracy, but it also makes the results less dependent on the image resolution. Of course the resolution still plays a role though, since less information is encoded in a lower resolution image. Thus we still get a limited accuracy of our point correspondences.

One factor that might limit the application span of our method is the requirement of knowing the translations and an initial guess for the rotations, but for the scope of this thesis and for Zenseact the translations are assumed to be known. Also, most car manufacturers probably have CAD models that specify how the cameras are mounted on their cars. In addition, both the articles [13] and [12] also assume a known initial pose estimate. If for any reason someone would want to implement our method on a problem where the translations are unknown or change with time, for example on a slowly deformable body, then they would need some way of estimating the poses first.

6

Conclusion

In this thesis a method for online correction of camera rotations was developed. Using point correspondences between overlapping fields of view, the rotations are optimized simultaneously starting from the ideal camera poses, here taken from the CAD model of the car. The method works in a wide range of environments, not requiring any special objects or landmarks. The only limitation is that we only make use of point correspondences from objects within a distance of 20 m from the cameras. This implies that the method might work poorly in wide open spaces such as some highways or large parking spots. However, that is not a big issue since the driving environment typically changes regularly and the calibration is not usually urgently needed.

We used two evaluation methods. The first one is visual inspection of birds eye view images, created through projection of fisheye camera images onto the flat ground plane. Generating such 360° surround views around the car is also the main intended use case of the fisheye cameras. Looking at the alignment of e.g. shadows and lane markings on the ground plane in overlapping views between cameras, we can directly determine the quality of our camera poses. The second evaluation metric is the reprojection error for manually annotated point correspondences. The correspondences are triangulated to estimate 3D points, which are then projected back into the images. The pixel distance between annotated and reprojected points indicate the accuracy of the extrinsic calibration. Well aligned birds eye view images seem to correlate with a low annotation reprojection error, so together these evaluation methods should reflect the accuracy of a given calibration.

The results indicate that the method achieves high accuracy extrinsic calibrations. For both evaluation criteria, our method outperforms both the factory calibration (that was done some time ago, i.e. not a fresh calibration) and the motion estimation method in [1], evaluated over scenarios containing only fisheye images. For scenarios with all cameras included, we also seem to get better accuracy with our method. Comparing to the dynamic pattern rig calibration, we achieve similar accuracy for both the scenarios with only fisheye cameras and the ones with all cameras. This is very promising, since the pattern calibration is tedious and requires quite a large designated indoor space. A fully automated online calibration that runs while driving the car is much more convenient. We also compared the relative angles between

our calibrations to the pattern based calibrations for scenarios with relatively recent calibrations. The deviation seems to typically be on the order of $0.1° - 0.4°$. The exact deviation is unattainable, since we do not have a ground truth to compare against. The accuracy is comparable to other state of the art methods for calibration of the fisheye camera rig, but we have also shown a proof of concept where other cameras may be included in the calibration as well.

In future work, there are a few things that need to be altered or investigated, especially if the method were to be implemented in a real car. Firstly, one would probably need to develop an online evaluation criterion that does not require human input. This should be used to avoid having a bad calibration affecting the decision making software running on the car. Secondly, for the case of optimizing all cameras, the timestamp difference is a problem. One solution might be to only collect images when the car is stationary. This would limit the generality of the method, but should contribute to an increased accuracy. Finally, one might want to investigate how accurately the translations and intrinsic parameters of cameras are known, and whether there is a way to increase their accuracy.

Bibliography

[1] Amit Dekel, Linus Härenstam-Nielsen, and Sergio Caccamo, "Optimal least-squares solution to the hand-eye calibration problem," *CoRR*, 2020. DOI: https://doi.org/10.48550/arXiv.2002.10838.

[2] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd edition. Cambridge University Press, Mar. 2011, p. 655, ISBN: 9780521540513.

[3] J. Kannala and S. S. Brandt, "A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses," Tech. Rep., 2006. DOI: 10.1109/TPAMI.2006.153.

[4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. DOI: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[5] X. Zhao, X. Wu, J. Miao, W. Chen, P. C. Y. Chen, and Z. Li, "ALIKE: Accurate and Lightweight Keypoint Detection and Descriptor Extraction," *IEEE Transactions on Multimedia*, vol. 25, pp. 3101–3112, 2023. DOI: 10.1109/TMM.2022.3155927.

[6] X. Zhao, X. Wu, W. Chen, P. C. Y. Chen, Q. Xu, and Z. Li, "ALIKED: A Lighter Keypoint and Descriptor Extraction Network via Deformable Transformation," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, 2023. DOI: 10.1109/TIM.2023.3271000.

[7] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local Feature Matching at Light Speed," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Jun. 2023, pp. 17 581–17 592. DOI: https://doi.org/10.48550/arXiv.2306.13643.

[8] S. Zhang and J. Ma, "DiffGlue: Diffusion-Aided Image Feature Matching," in *MM 2024 - Proceedings of the 32nd ACM International Conference on Multimedia*, Association for Computing Machinery, Inc, Oct. 2024, pp. 8451–8460, ISBN: 9798400706868. DOI: 10.1145/3664647.3681069.

[9] H. Jiang, A. Karpur, B. Cao, Q. Huang, and A. Araujo, "OmniGlue: Generalizable Feature Matching with Foundation Model Guidance," May 2024. DOI: 10.1109/CVPR52733.2024.01878.

[10] J. Zhao Tusimple and L. Kneip, "Hybrid Rotation Averaging: A Fast and Robust Rotation Averaging Approach," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 353–10 362. DOI: 10.1109/CVPR46437.2021.01022.

[11] G. Gallego and A. Yezzi, "A Compact Formula for the Derivative of a 3-D Rotation in Exponential Coordinates," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 3, pp. 378–384, Mar. 2015, ISSN: 15737683. DOI: 10.1007/s10851-014-0528-x.

[12] T. Zhang, N. Zhao, Y. Shen, X. Shao, L. Zhang, and Y. Zhou, "ROECS: A Robust Semi-direct Pipeline towards Online Extrinsics Correction of the Surround-view System," in *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, Association for Computing Machinery, Inc, Oct. 2021, pp. 3153–3161, ISBN: 9781450386517. DOI: 10.1145/3474085.3475461.

[13] L. Wang, *Click-Calib: A Robust Extrinsic Calibration Method for Surround-View Systems*, Jan. 2025. DOI: https://doi.org/10.48550/arXiv.2501.01557.

[14] M. Knorr, J. Esparza, W. Niehsen, and C. Stiller, "Extrinsic calibration of a fisheye multi-camera setup using overlapping fields of view," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, IEEE, Jun. 2014, pp. 1276–1281, ISBN: 978-1-4799-3638-0. DOI: 10.1109/IVS.2014.6856403.

[15] G. Carrera, A. Angeli, and A. J. Davison, "SLAM-Based Automatic Extrinsic Calibration of a Multi-Camera Rig," Tech. Rep., 2011. DOI: 10.1109/ICRA.2011.5980294. [Online]. Available: https://ieeexplore.ieee.org/document/5980294.

[16] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic Intrinsic and Extrinsic Calibration of a Rig with Multiple Generic Cameras and Odometry," Tech. Rep., 2013. DOI: 10.1109/IROS.2013.6696592.

[17] K. Choi, H. G. Jung, and J. K. Suhr, "Automatic calibration of an around view monitor system exploiting lane markings," *Sensors (Switzerland)*, vol. 18, no. 9, Sep. 2018, ISSN: 14248220. DOI: 10.3390/s18092956.

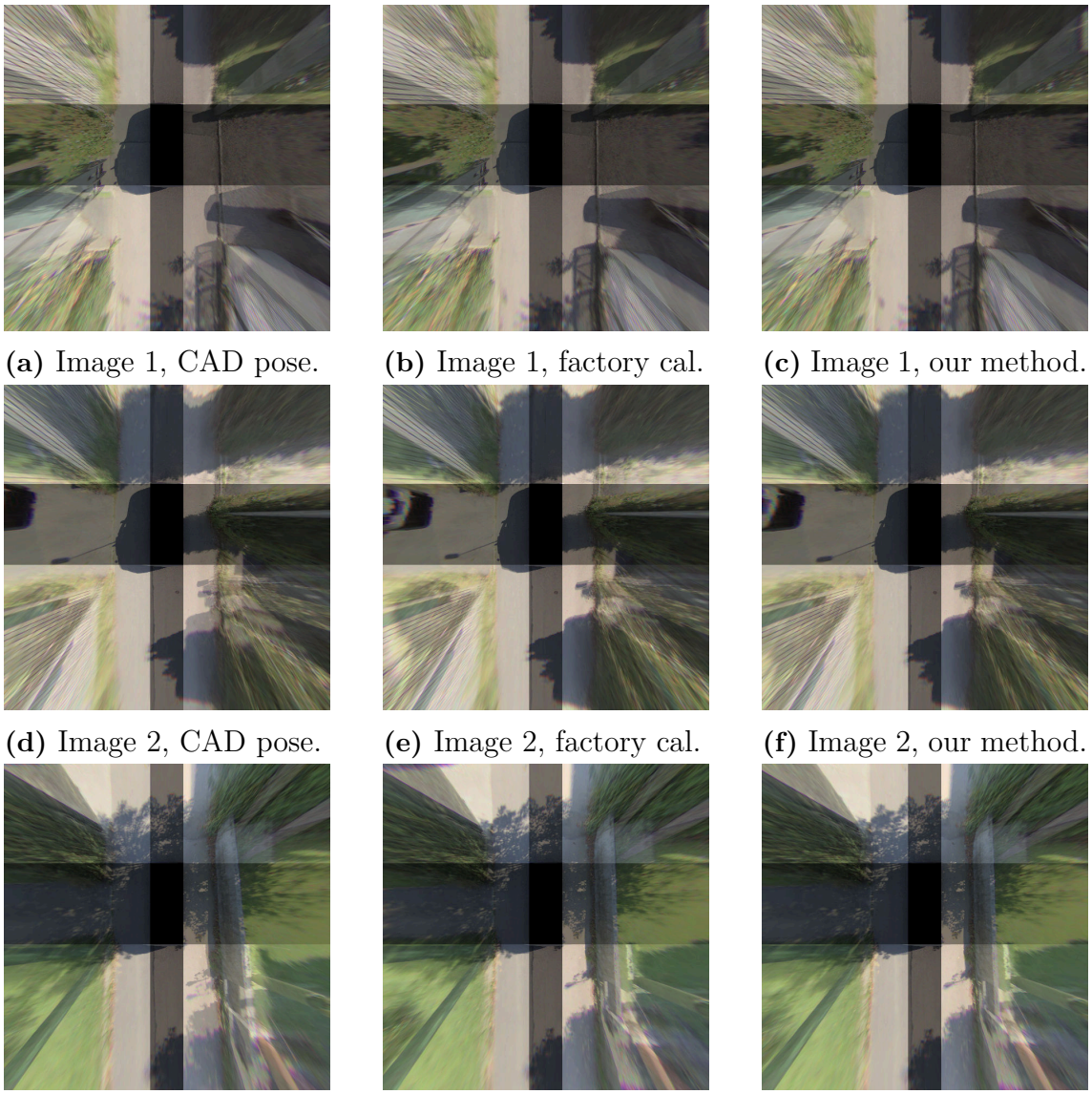
[18] A. E. Beaton and J. W. Tukey, "The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data," *Technometrics*, vol. 16, no. 2, pp. 147–185, May 1974, ISSN: 0040-1706. DOI: 10.1080/00401706.1974.10489171.

[19] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust Optimization for Deep Regression," Tech. Rep., 2015. DOI: 10.1109/ICCV.2015.324.

A

Appendix 1

The birds eye view images for the scenarios with all cameras are shown on the following pages.

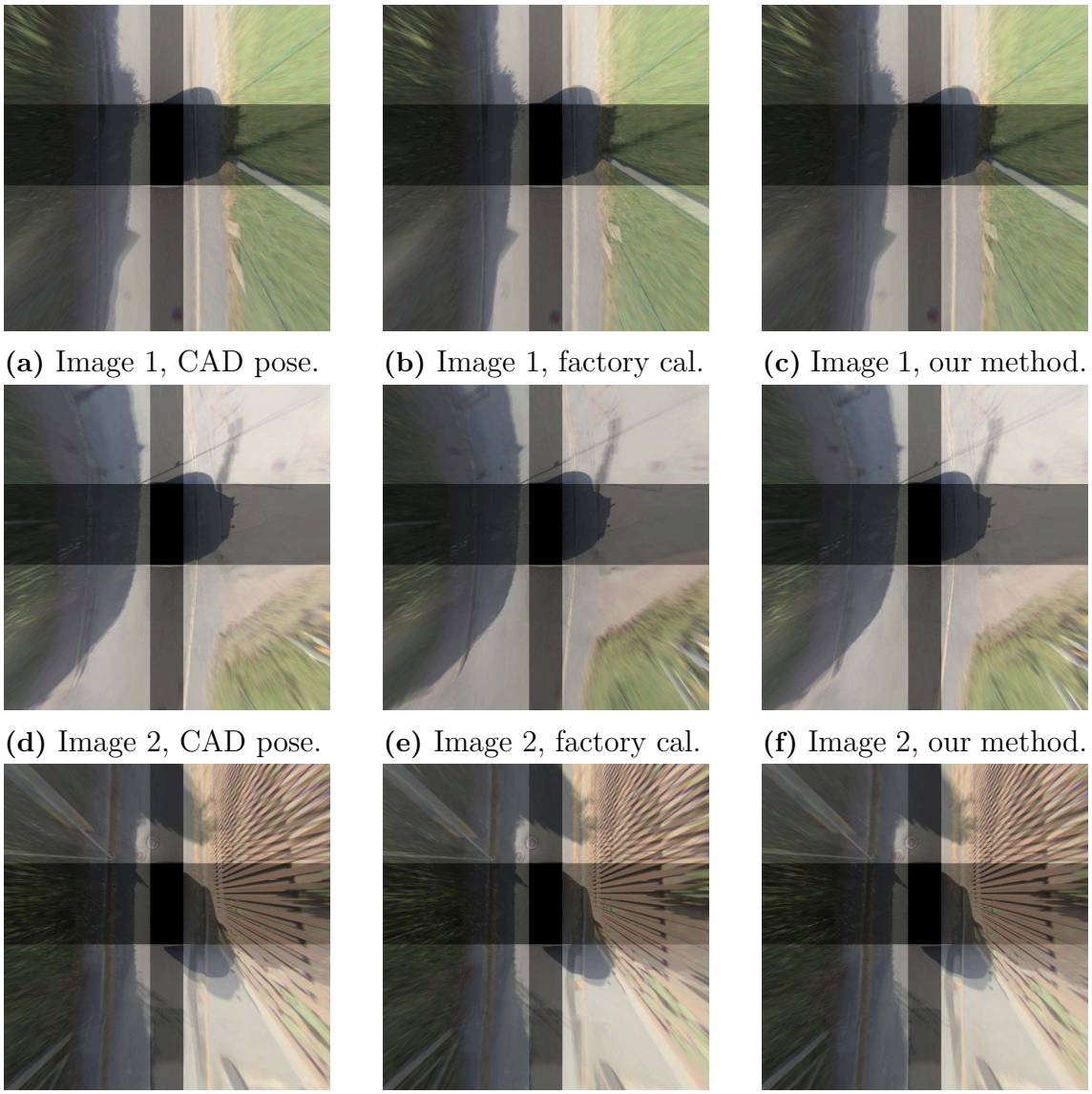


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.1: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 21 with all cameras included in the optimization.

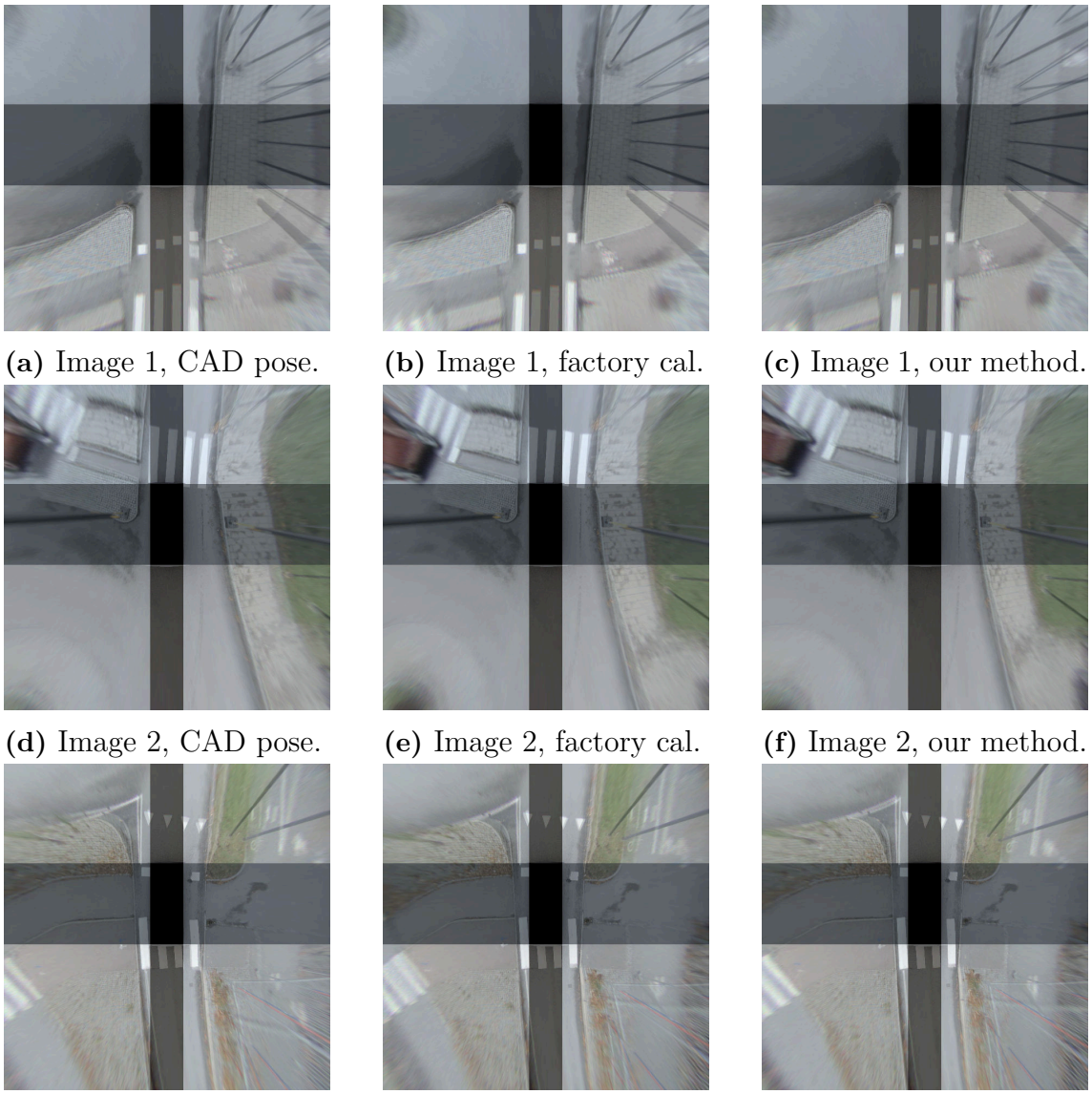


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.2: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 21 with all cameras included in the optimization.

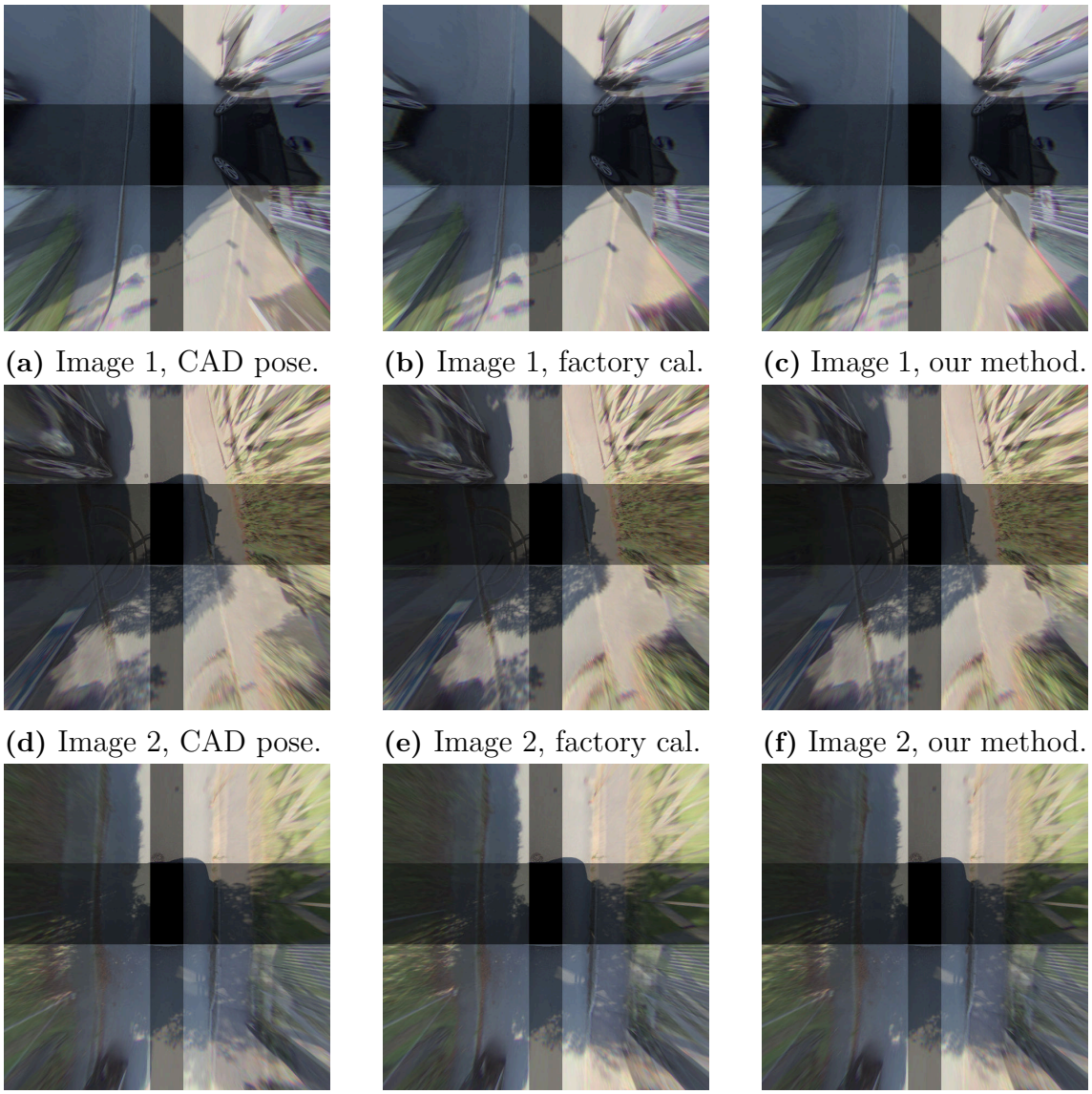


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.3: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 21 with all cameras included in the optimization.

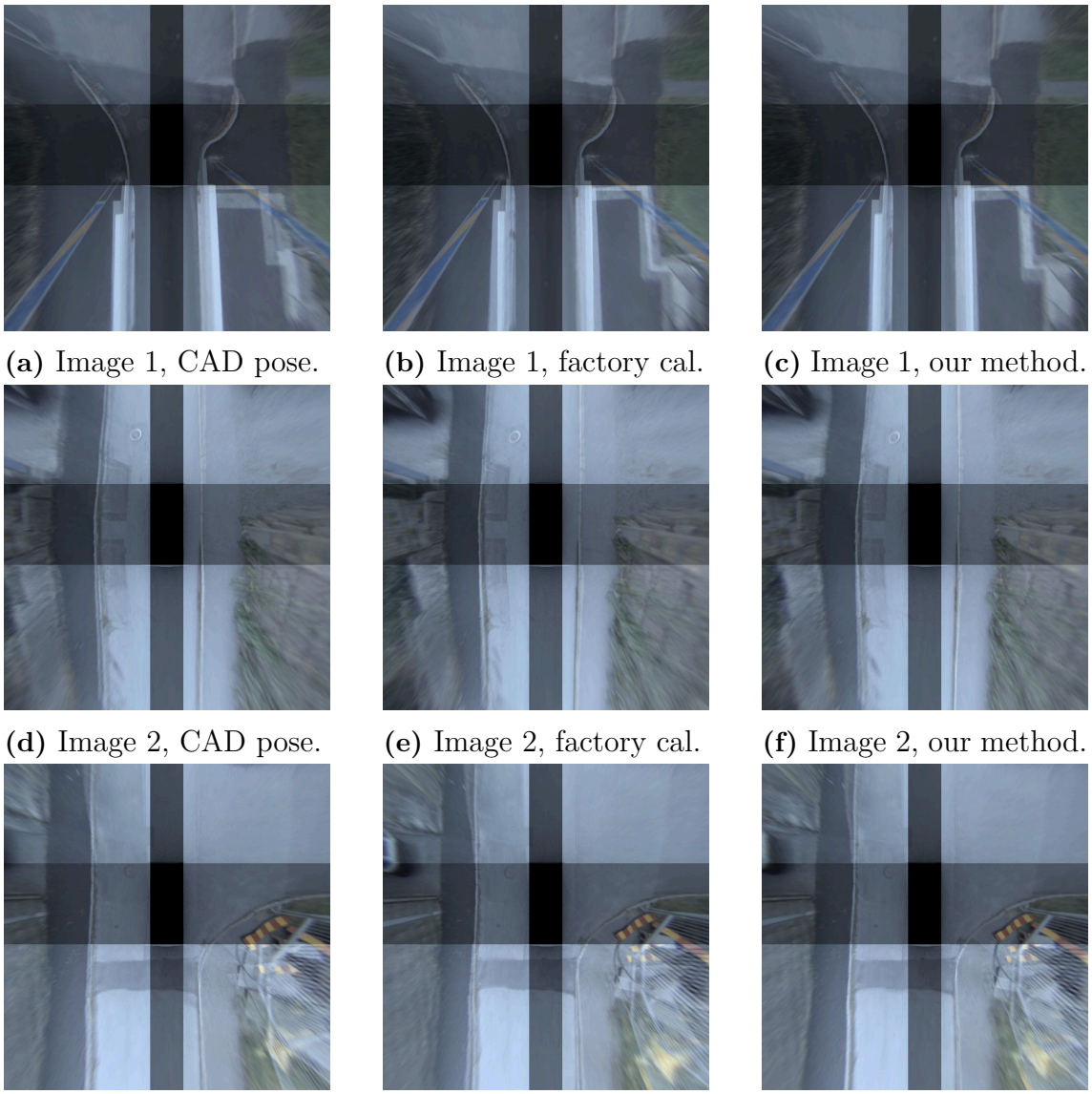


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.4: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 21 with all cameras included in the optimization.

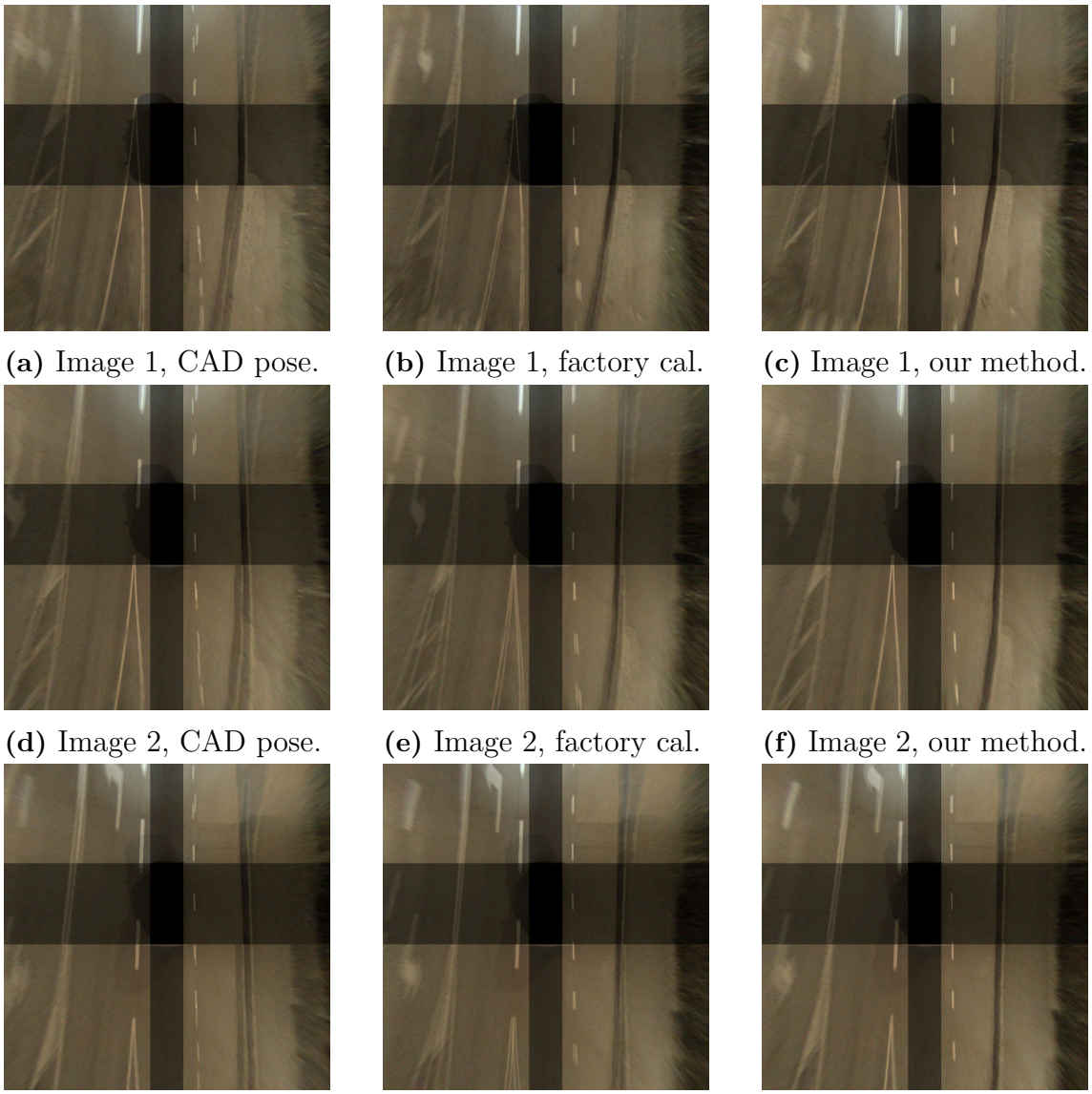


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.5: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 25 with all cameras included in the optimization.

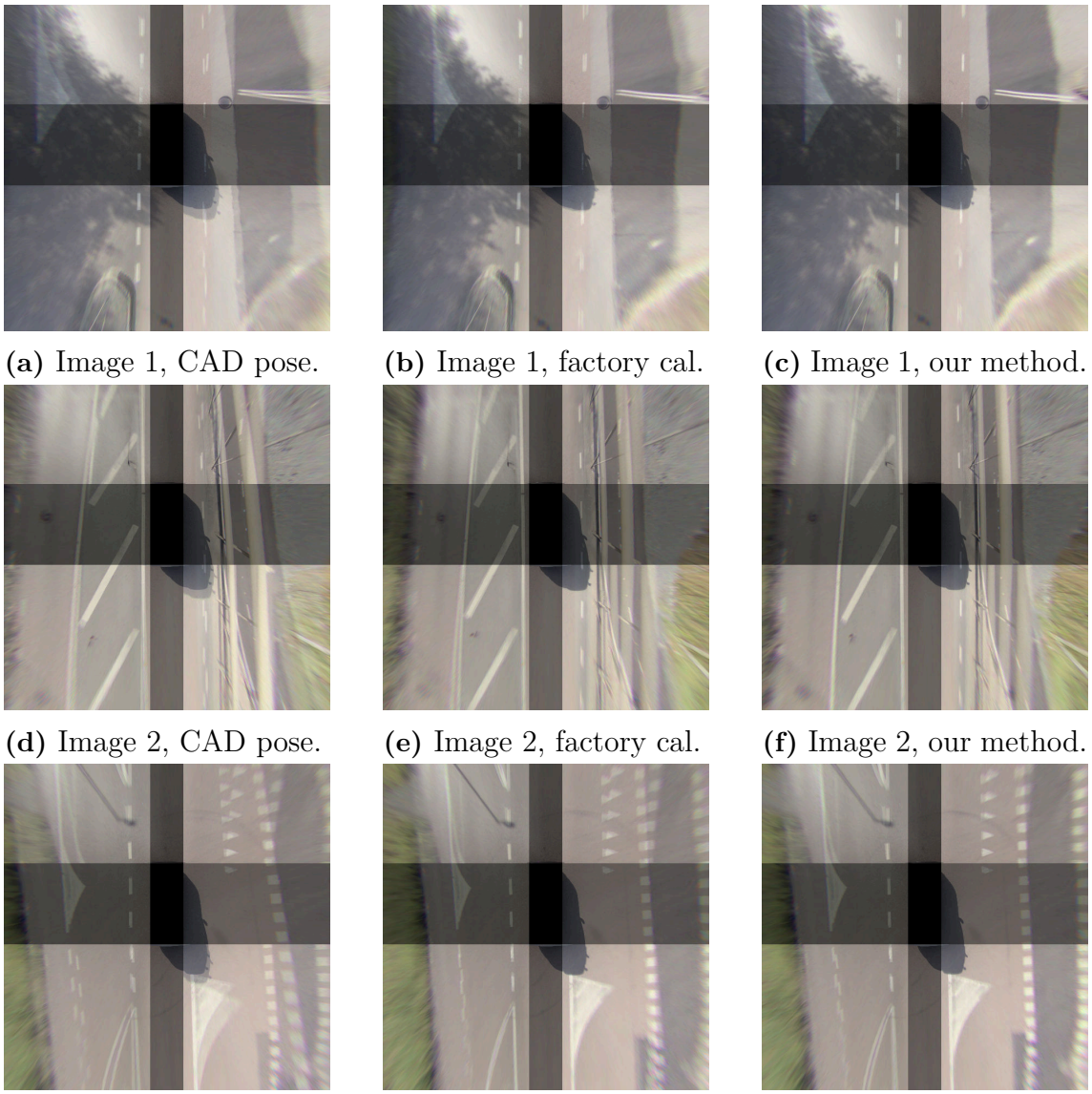


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.6: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 26 with all cameras included in the optimization.

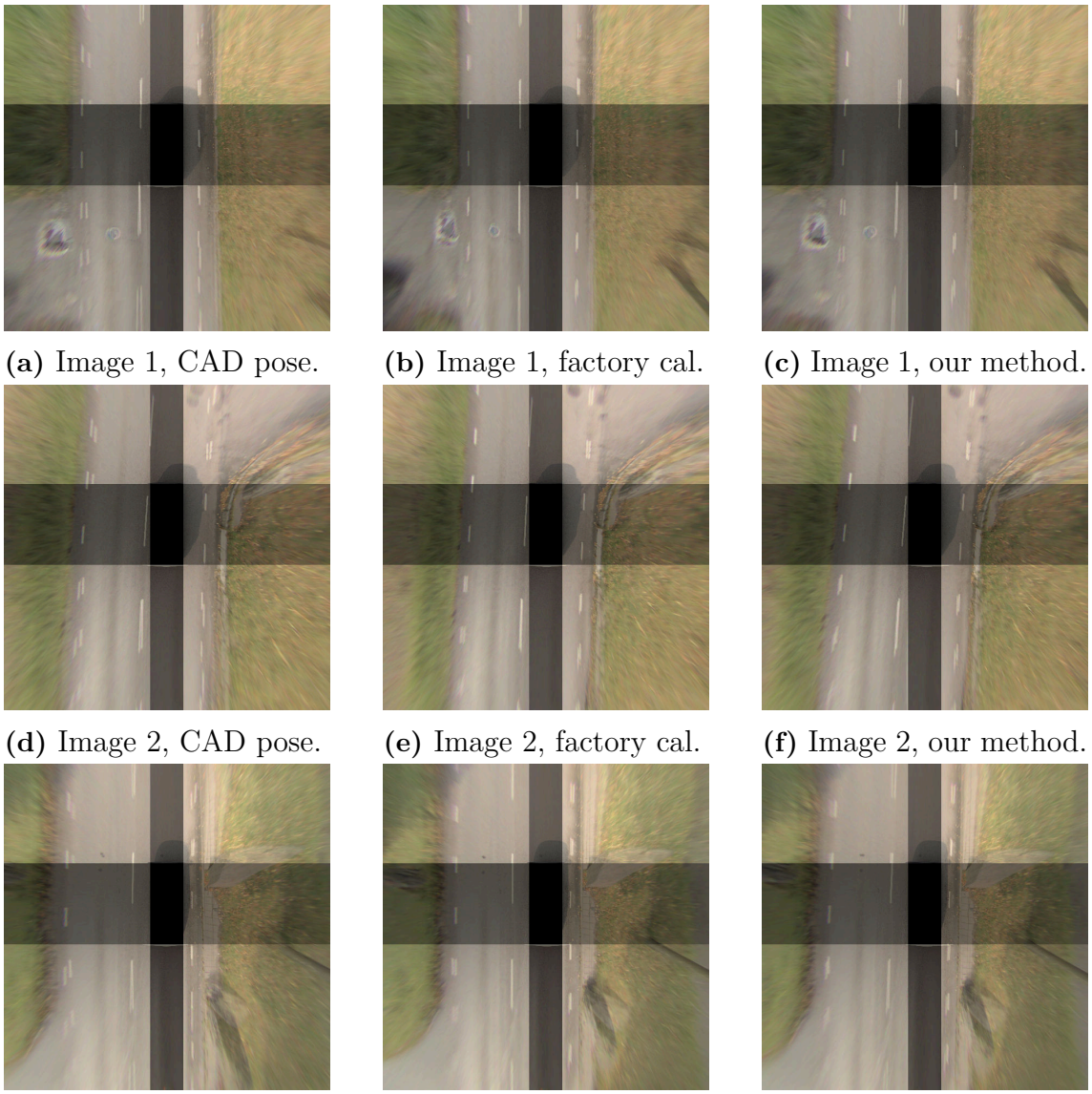


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.7: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 27 with all cameras included in the optimization.

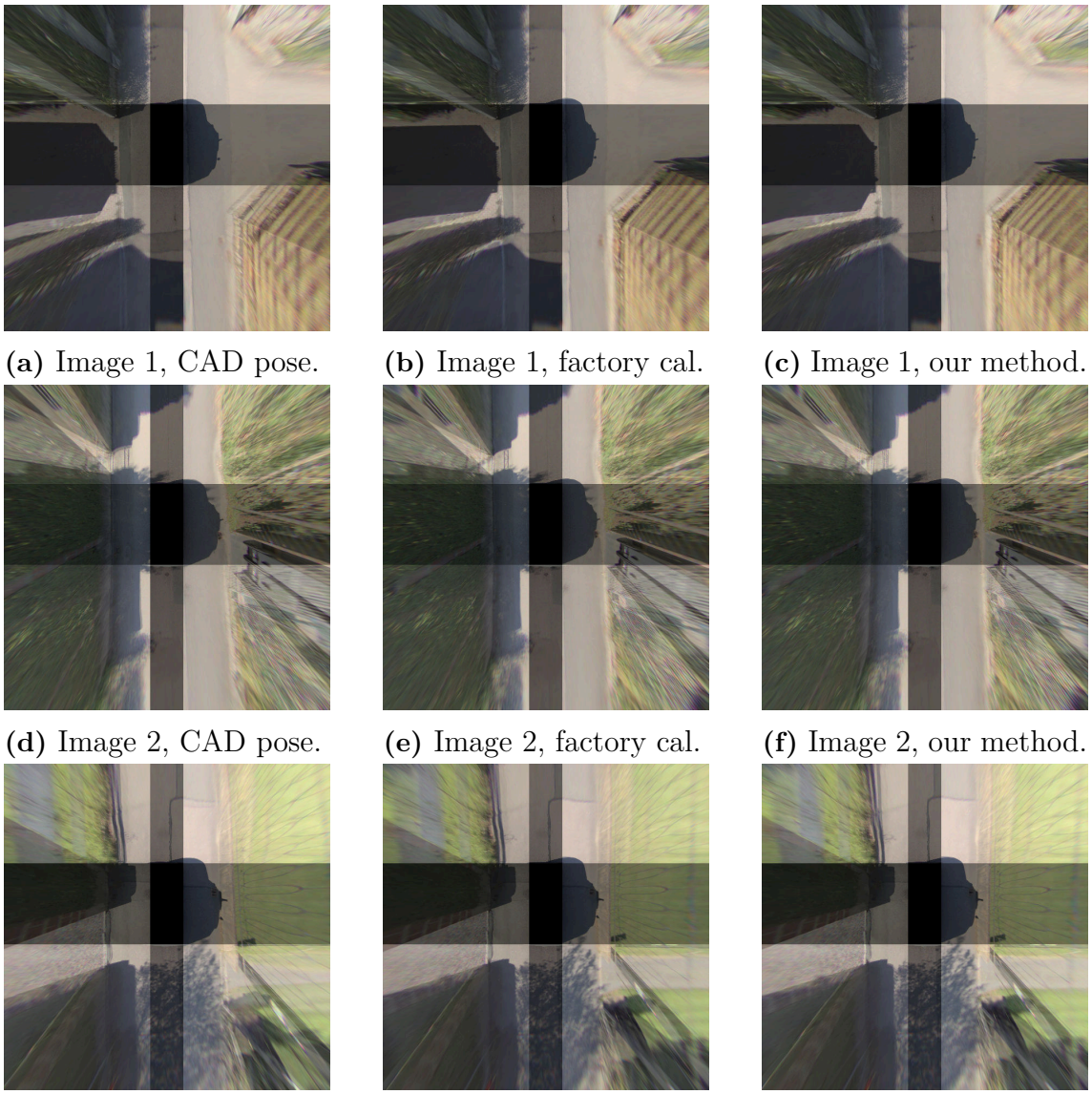


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.8: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 28 with all cameras included in the optimization.

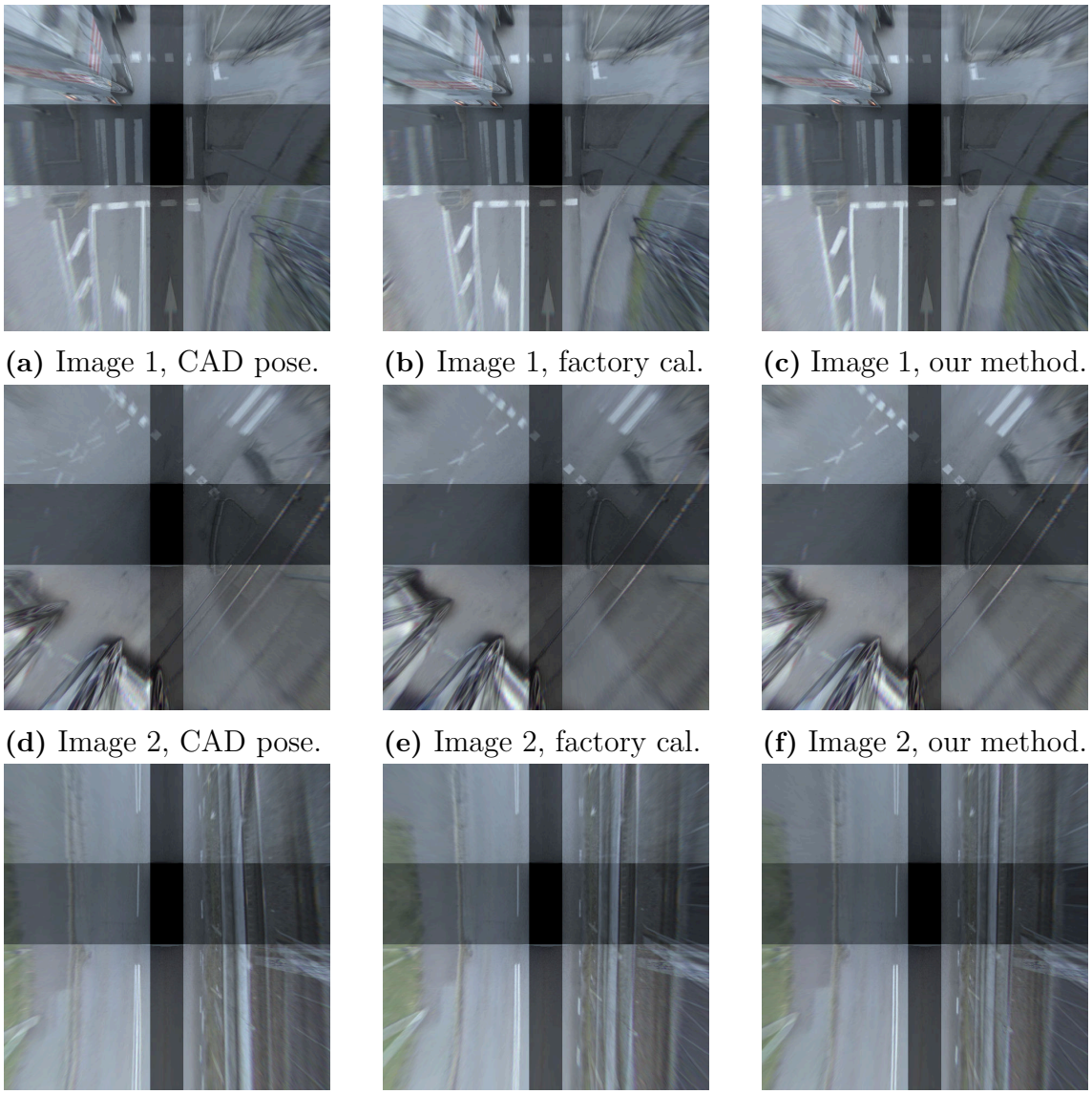


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.9: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 29 with all cameras included in the optimization.

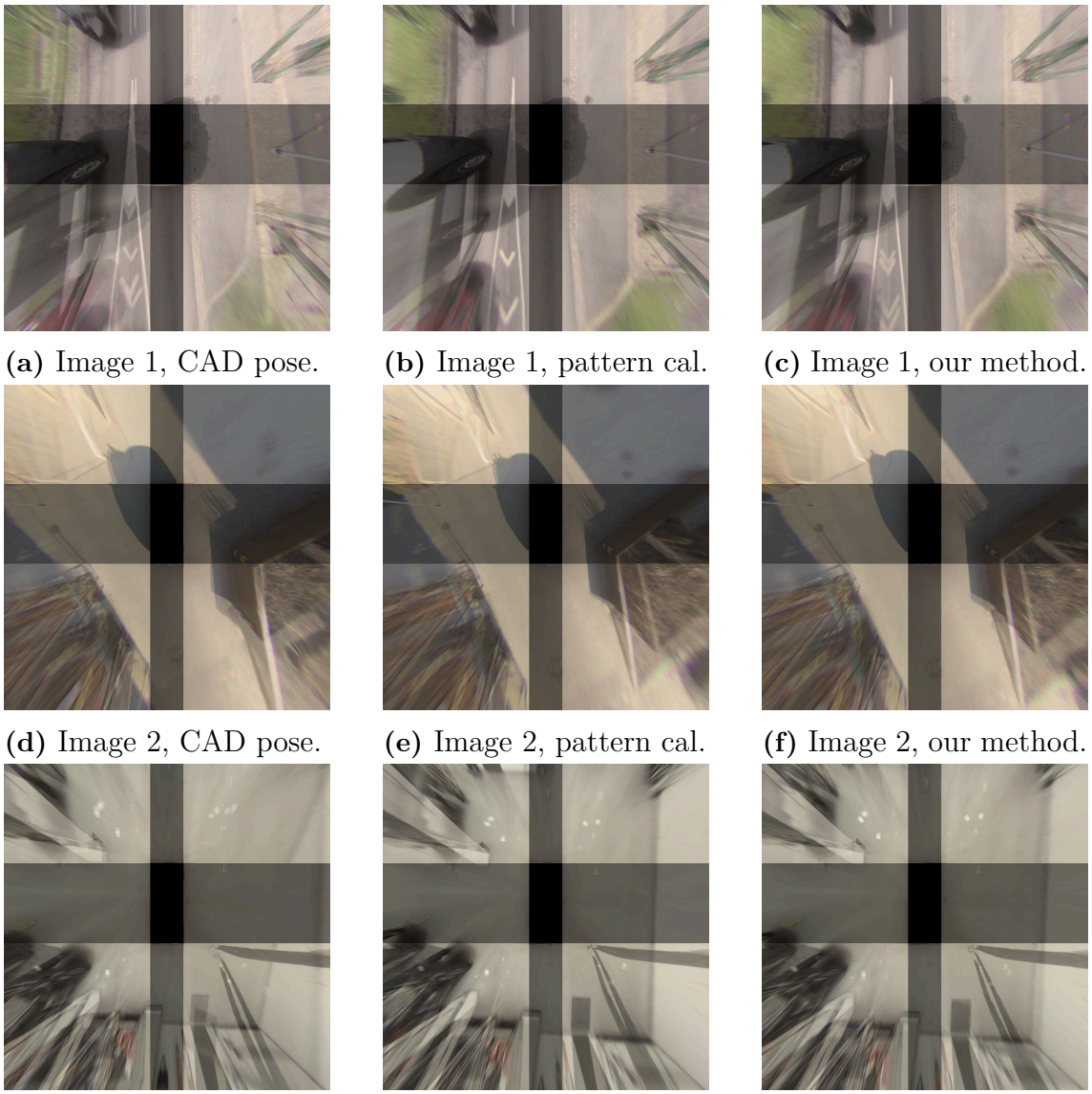


(a) Image 1, CAD pose. **(b)** Image 1, factory cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, factory cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, factory cal. **(i)** Image 3, our method.

Figure A.10: Birds eye view images created using the poses from the CAD model, the factory calibration and our optimization method for scenario 30 with all cameras included in the optimization.

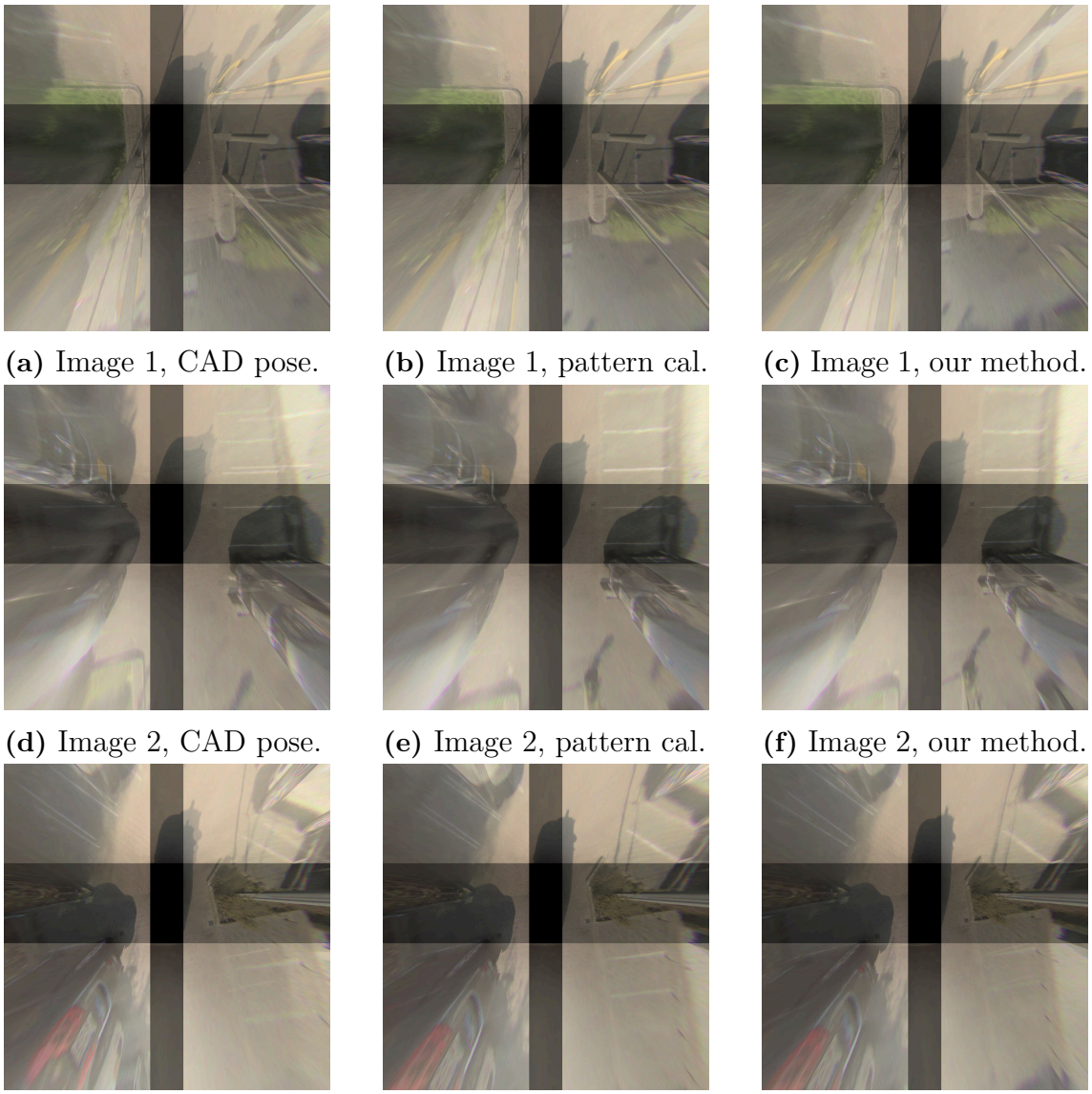


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.11: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 31 with all cameras included in the optimization.

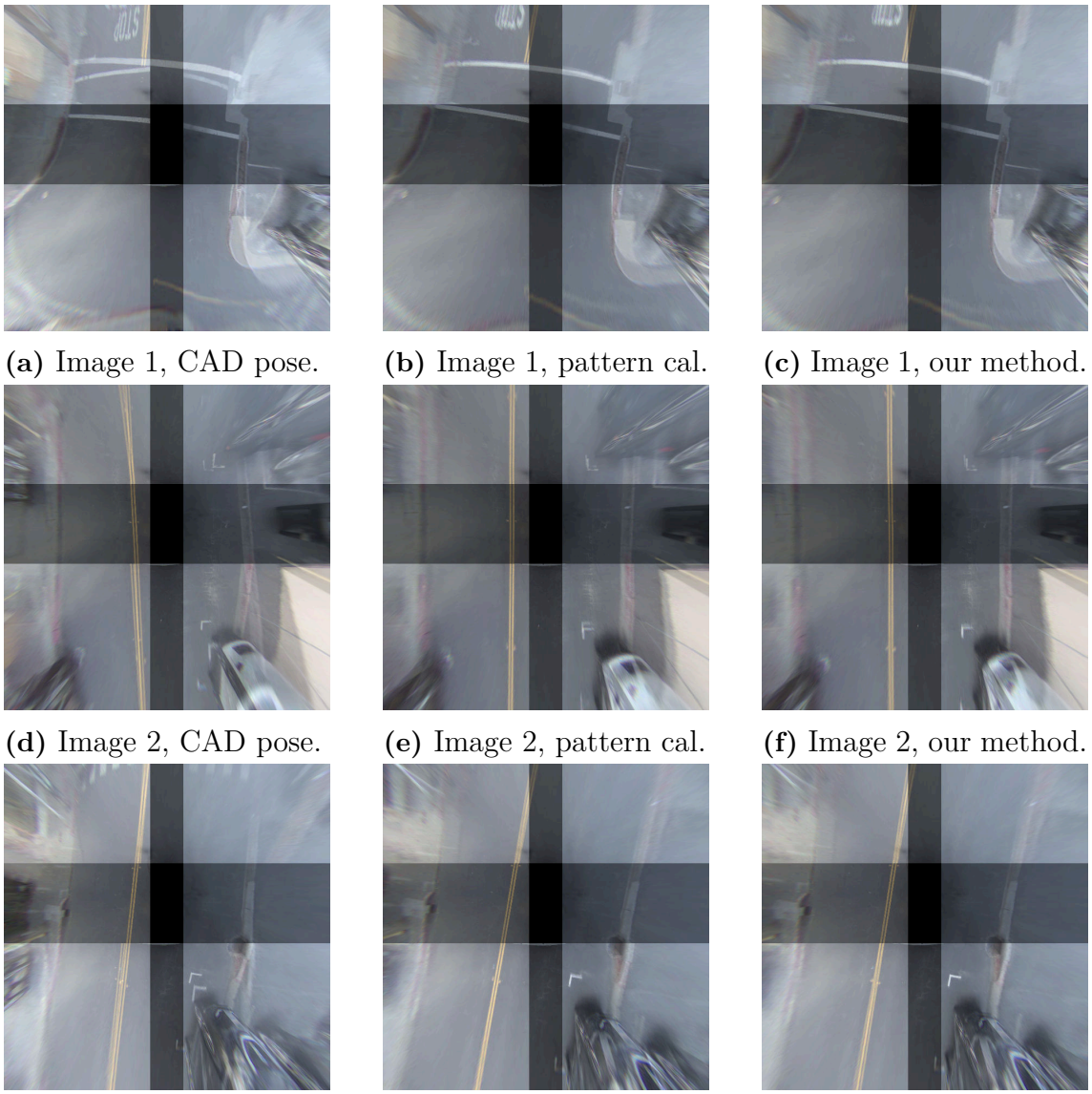


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.12: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 32 with all cameras included in the optimization.

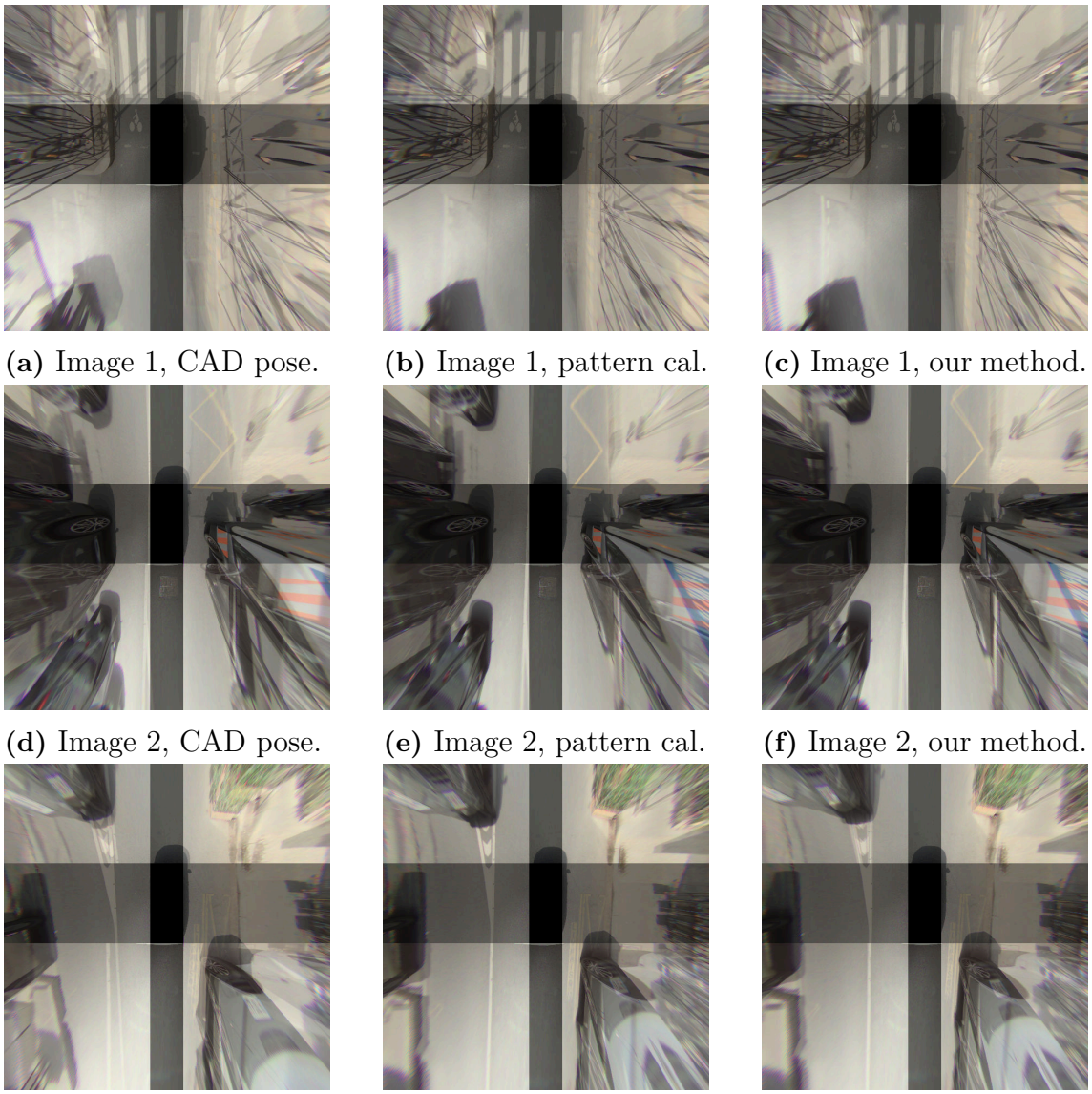


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.13: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 33 with all cameras included in the optimization.

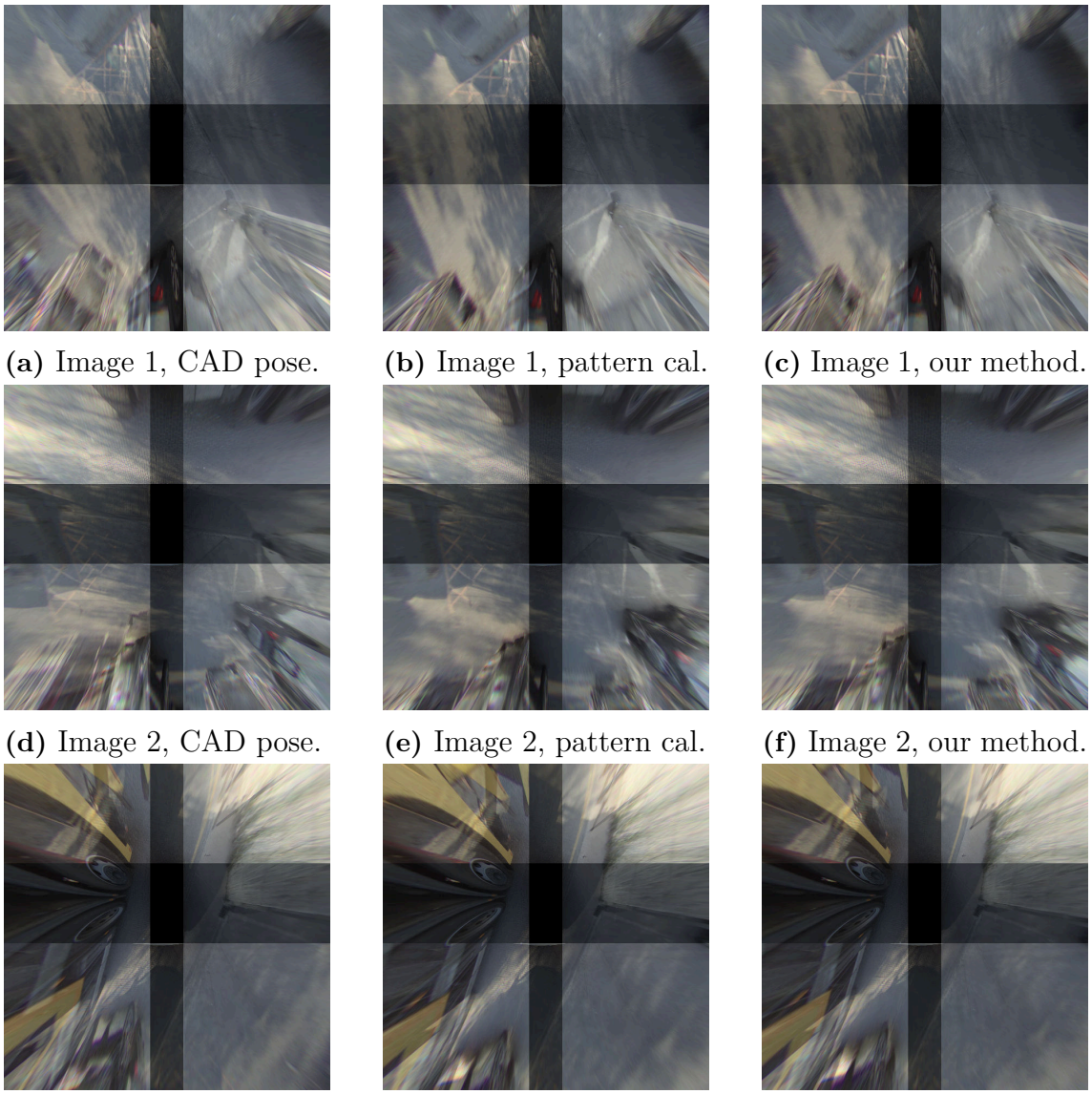


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.14: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 34 with all cameras included in the optimization.

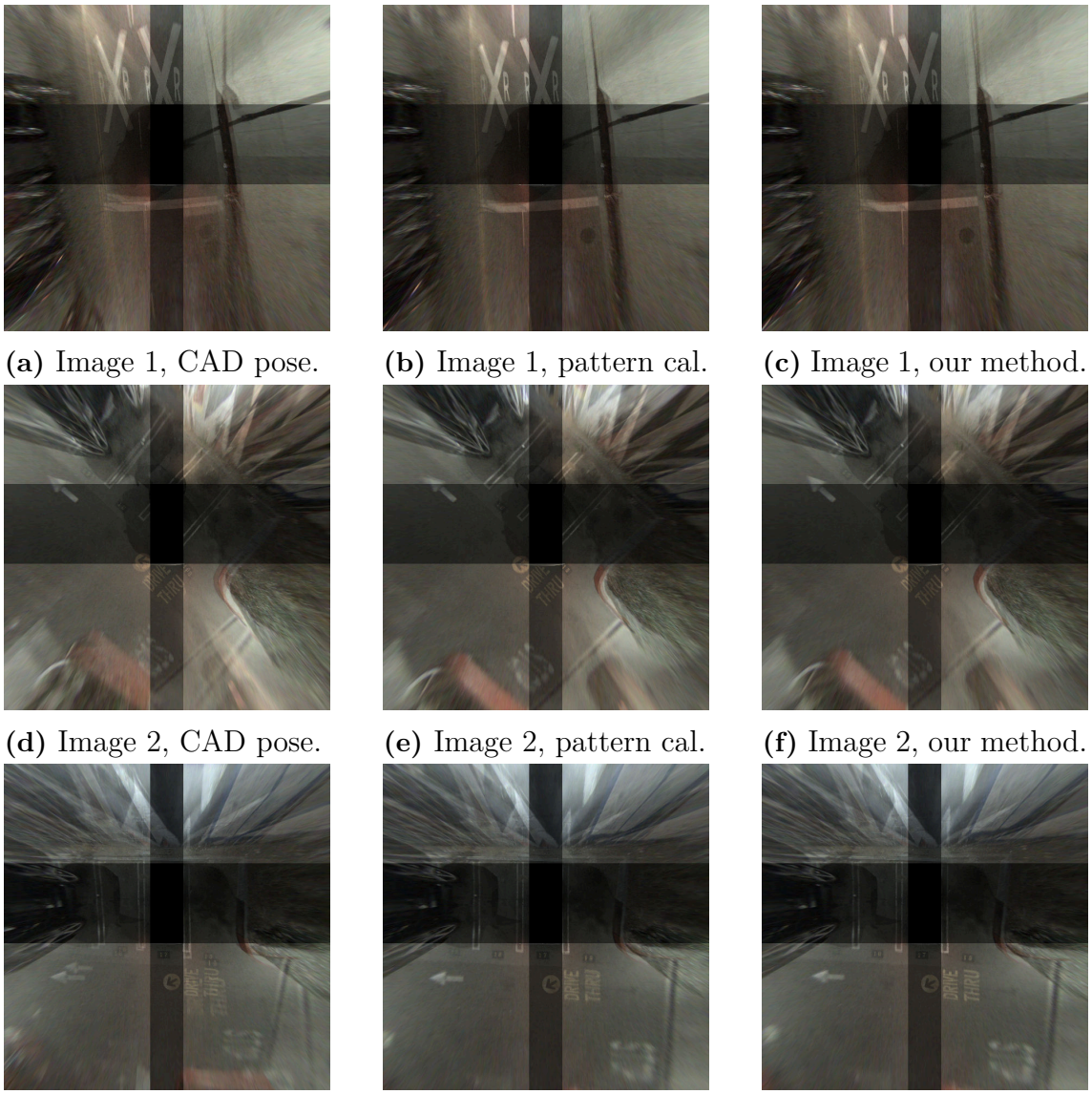


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.15: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 35 with all cameras included in the optimization.

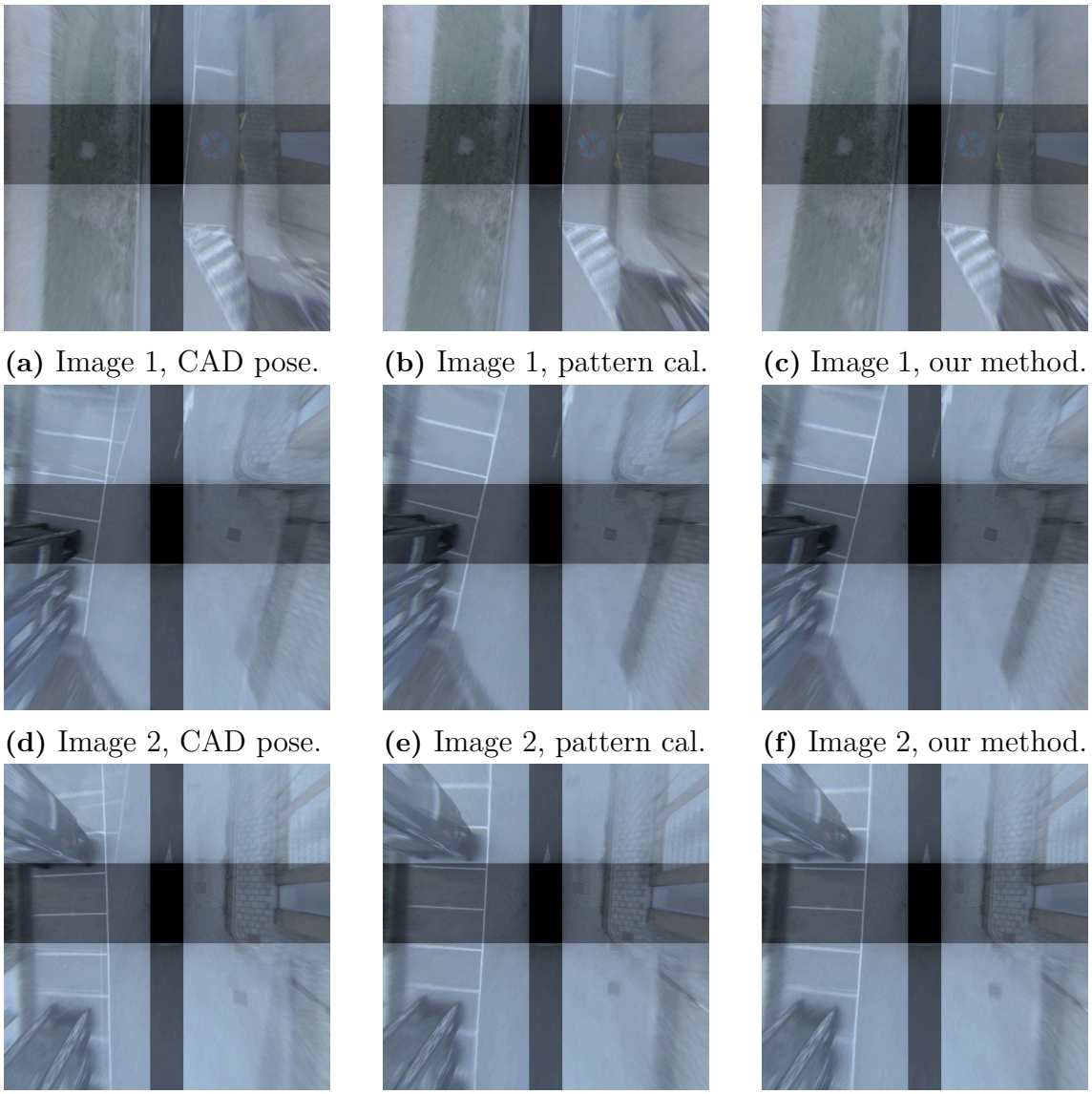


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.16: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 36 with all cameras included in the optimization.

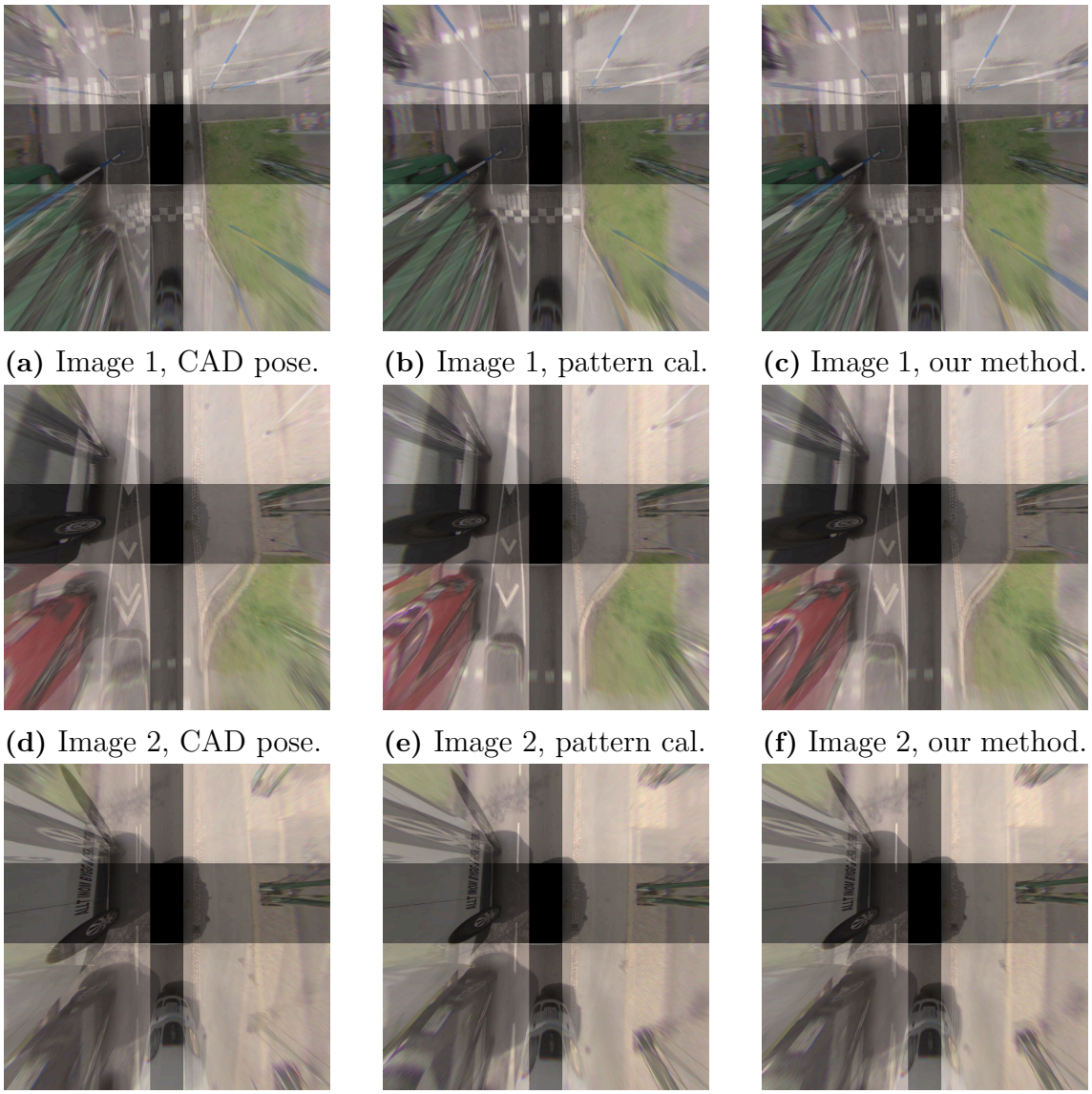


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.17: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 37 with all cameras included in the optimization.

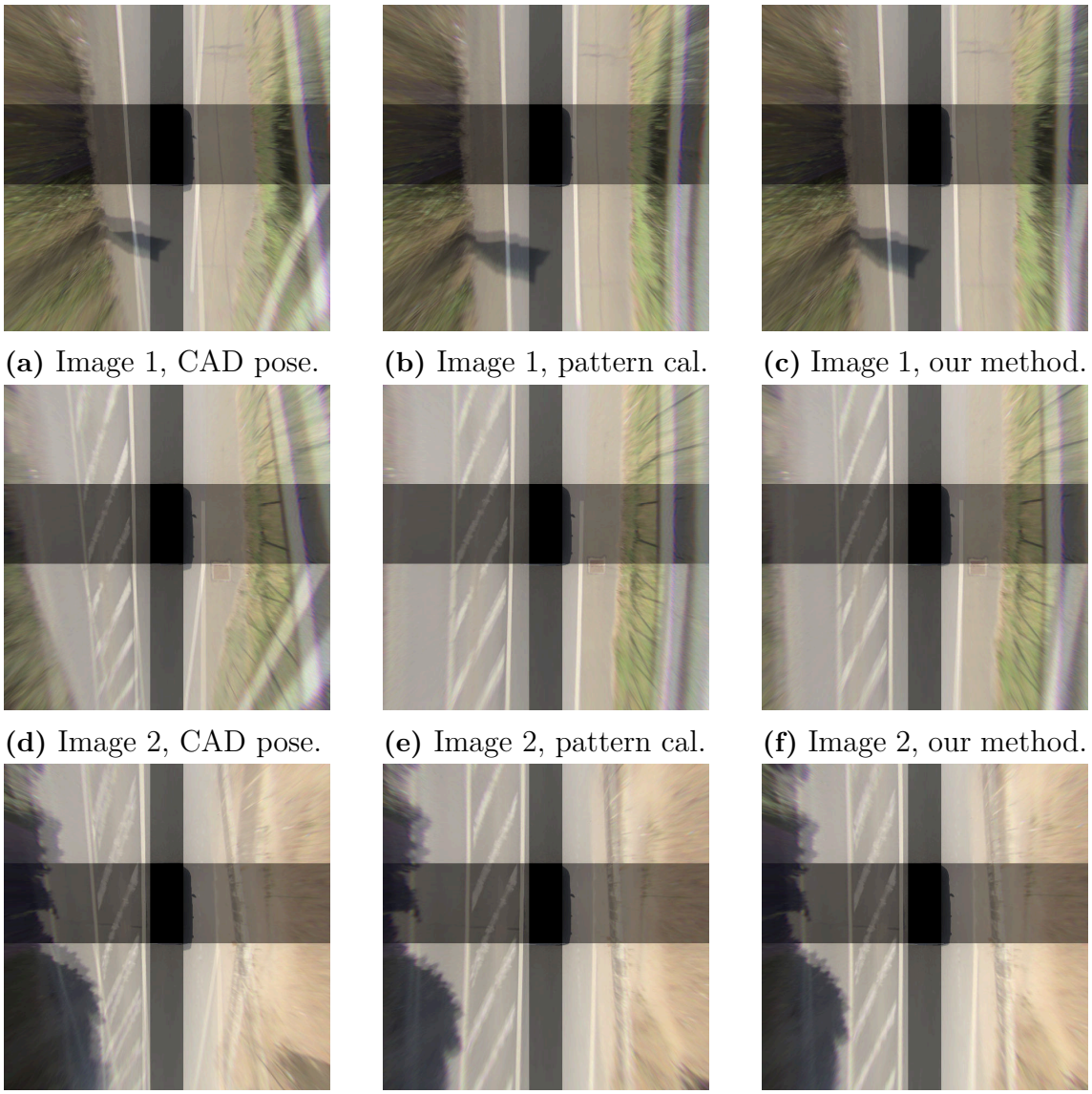


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.18: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 38 with all cameras included in the optimization.

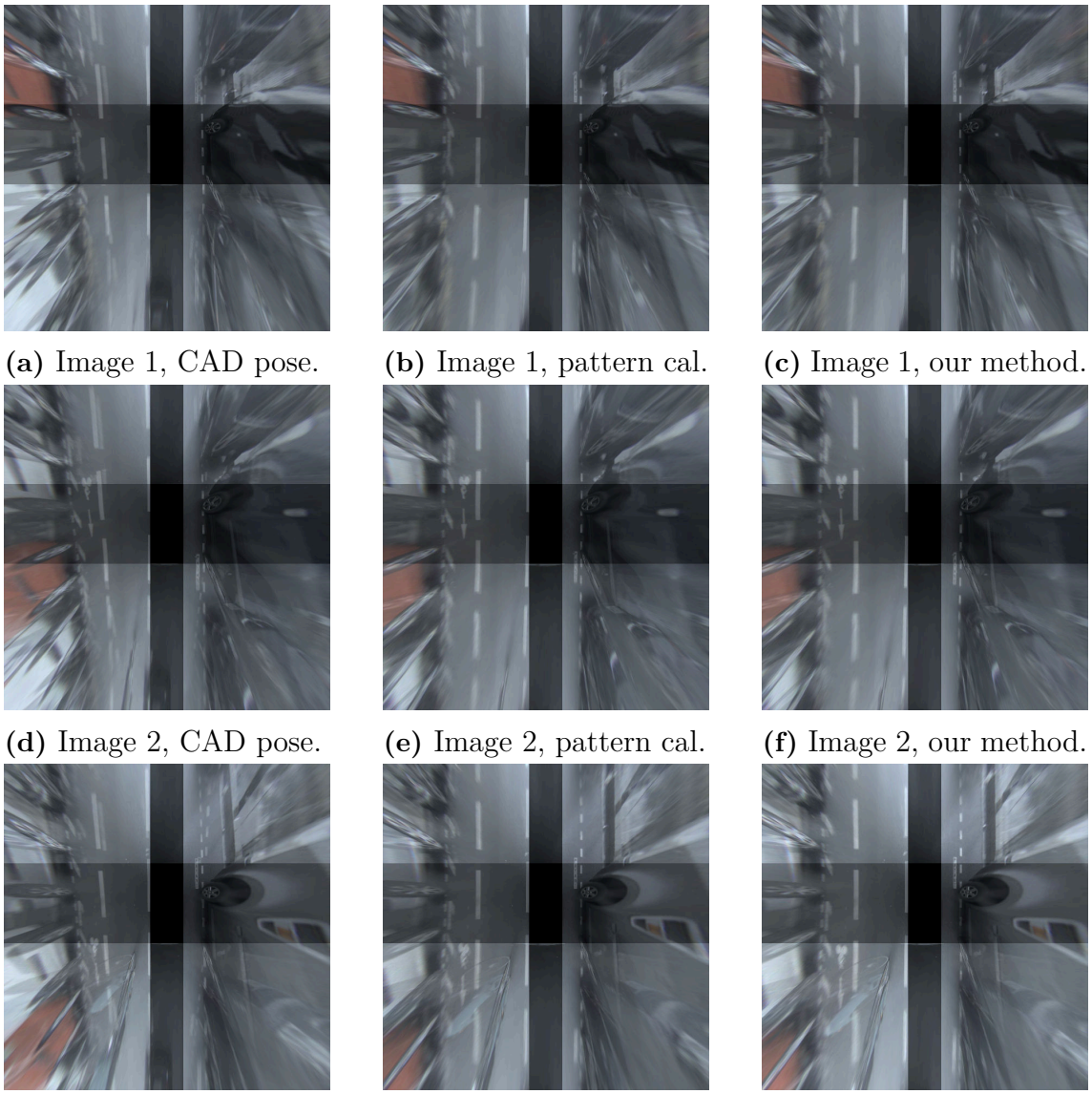


(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.19: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 39 with all cameras included in the optimization.



(a) Image 1, CAD pose. **(b)** Image 1, pattern cal. **(c)** Image 1, our method.

(d) Image 2, CAD pose. **(e)** Image 2, pattern cal. **(f)** Image 2, our method.

(g) Image 3, CAD pose. **(h)** Image 3, pattern cal. **(i)** Image 3, our method.

Figure A.20: Birds eye view images created using the poses from the CAD model, the latest dynamic pattern calibration and our optimization method for scenario 40 with all cameras included in the optimization.

B

Appendix 2

The following pages show the relative angles between the factory calibration or the dynamic pattern calibration (only one of them is available for each scenario). Note that these calibrations were done some time ago when the data was collected. The angles are given for the relative rotation matrix R_{rel} relating the factory/dynamic pattern calibration R_{pattern} to our calibration R_{our} by $R_{\text{pattern}} = R_{\text{rel}} R_{\text{our}}$.

Table B.1: Relative angles in degrees between the poses from our method and the factory calibration for scenario 1.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.59	-0.3	0.13
Front fisheye	1.37	0.54	-0.27
Right fisheye	0.89	-0.19	-0.4
Back fisheye	0.76	0.15	-0.22

Table B.2: Relative angles in degrees between the poses from our method and the factory calibration for scenario 2.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.91	-0.4	1.0
Front fisheye	-0.51	-0.36	0.9
Right fisheye	-0.21	-0.26	0.95
Back fisheye	-0.42	-0.11	0.87

Table B.3: Relative angles in degrees between the poses from our method and the factory calibration for scenario 3.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.94	-0.18	0.83
Front fisheye	-0.53	-0.29	1.02
Right fisheye	-0.46	-0.09	0.4
Back fisheye	-0.23	0.59	-0.25

Table B.4: Relative angles in degrees between the poses from our method and the factory calibration for scenario 4.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.79	0.06	1.21
Front fisheye	1.28	0.74	0.71
Right fisheye	-0.59	-0.07	0.16
Back fisheye	-0.35	0.46	0.68

Table B.5: Relative angles in degrees between the poses from our method and the factory calibration for scenario 5.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.02	-0.17	0.63
Front fisheye	1.73	0.44	-0.1
Right fisheye	0.24	-0.13	0.65
Back fisheye	0.23	0.32	0.71

Table B.6: Relative angles in degrees between the poses from our method and the factory calibration for scenario 6.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.5	-0.29	0.75
Front fisheye	0.02	0.24	0.66
Right fisheye	0.32	0.02	0.72
Back fisheye	-0.11	-0.16	0.75

Table B.7: Relative angles in degrees between the poses from our method and the factory calibration for scenario 7.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.06	-0.28	0.45
Front fisheye	0.85	0.55	-0.11
Right fisheye	0.42	-0.09	-0.08
Back fisheye	0.25	0.51	-0.26

Table B.8: Relative angles in degrees between the poses from our method and the factory calibration for scenario 8.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.24	-0.29	0.6
Front fisheye	0.77	0.65	-0.42
Right fisheye	0.11	-0.09	-0.08
Back fisheye	-0.08	0.33	0.81

Table B.9: Relative angles in degrees between the poses from our method and the factory calibration for scenario 9.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.26	-0.25	0.06
Front fisheye	1.08	0.69	-0.62
Right fisheye	0.67	-0.13	-0.62
Back fisheye	0.51	0.14	-0.36

Table B.10: Relative angles in degrees between the poses from our method and the factory calibration for scenario 10.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.37	-0.38	0.68
Front fisheye	1.46	0.4	-0.3
Right fisheye	-0.31	-0.04	-0.09
Back fisheye	-0.21	0.63	-0.08

Table B.11: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 11.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.01	-0.19	0.19
Front fisheye	-0.08	-0.08	0.57
Right fisheye	-0.01	0.04	0.51
Back fisheye	-0.17	0.22	1.17

Table B.12: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 12.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.25	-0.14	-0.39
Front fisheye	-0.19	0.02	-0.04
Right fisheye	-0.11	-0.22	-0.15
Back fisheye	-0.2	-0.09	-0.62

Table B.13: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 13.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.18	0.17	-1.08
Front fisheye	0.26	0.16	0.48
Right fisheye	0.37	0.29	-0.23
Back fisheye	0.02	0.15	-0.39

Table B.14: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 14.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.49	0.16	0.18
Front fisheye	-0.3	0.33	-0.05
Right fisheye	-0.05	0.42	0.59
Back fisheye	-0.23	0.52	0.85

Table B.15: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 15.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.35	-0.09	-0.39
Front fisheye	0.42	0.3	-0.57
Right fisheye	0.39	0.26	-0.49
Back fisheye	0.14	0.19	-0.19

Table B.16: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 16.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.1	-0.21	-0.95
Front fisheye	0.21	-0.38	-0.39
Right fisheye	0.18	0.06	-0.25
Back fisheye	-0.03	-0.11	-0.94

Table B.17: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 17.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.25	-0.1	0.03
Front fisheye	-0.34	0.14	0.02
Right fisheye	-0.15	0.35	-0.37
Back fisheye	-0.35	0.15	0.75

Table B.18: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 18.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.61	-0.18	-0.92
Front fisheye	0.4	-0.02	-1.48
Right fisheye	0.61	0.25	-0.1
Back fisheye	0.29	0.1	0.2

Table B.19: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 19.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.46	-0.16	-1.17
Front fisheye	0.24	-0.42	-1.54
Right fisheye	0.52	0.0	0.03
Back fisheye	0.4	-0.01	-0.43

Table B.20: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 20.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.39	0.14	-0.23
Front fisheye	0.13	0.18	-0.2
Right fisheye	0.73	0.24	0.32
Back fisheye	0.03	0.71	5.29

Table B.21: Relative angles in degrees between the poses from our method and the factory calibration for scenario 21.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.55	-0.15	0.14
Front fisheye	-0.24	0.17	0.13
Right fisheye	-0.11	0.11	0.15
Back fisheye	-0.19	0.07	-0.09
Long range front	-0.29	0.04	0.26
Long range back left	-0.43	0.09	-0.01
Long range back right	-0.06	0.48	0.3

Table B.22: Relative angles in degrees between the poses from our method and the factory calibration for scenario 22.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.52	-0.28	1.14
Front fisheye	0.12	-0.03	0.69
Right fisheye	0.5	-0.07	0.32
Back fisheye	-0.23	0.1	0.15
Long range front	0.22	-0.14	0.77
Long range back left	-0.13	-0.06	0.05
Long range back right	-0.03	0.36	0.4

Table B.23: Relative angles in degrees between the poses from our method and the factory calibration for scenario 23.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.56	-0.12	0.56
Front fisheye	-0.29	-0.15	0.39
Right fisheye	-0.18	0.07	0.37
Back fisheye	-0.21	0.41	0.09
Long range front	-0.27	-0.18	0.52
Long range back left	-0.45	0.3	0.06
Long range back right	-0.39	0.63	0.48

Table B.24: Relative angles in degrees between the poses from our method and the factory calibration for scenario 24.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.41	-0.22	0.61
Front fisheye	0.02	0.19	0.25
Right fisheye	0.24	0.15	0.43
Back fisheye	-0.08	0.07	-0.03
Long range front	0.19	0.02	0.41
Long range back left	-0.17	0.07	-0.08
Long range back right	0.06	0.47	0.32

Table B.25: Relative angles in degrees between the poses from our method and the factory calibration for scenario 25.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.42	-0.21	0.67
Front fisheye	-0.11	-0.22	0.28
Right fisheye	0.11	0.11	0.11
Back fisheye	-0.14	0.5	0.16
Long range front	-0.14	-0.2	0.36
Long range back left	-0.35	0.16	0.31
Long range back right	-0.17	0.25	0.21

Table B.26: Relative angles in degrees between the poses from our method and the factory calibration for scenario 26.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.43	-0.29	0.63
Front fisheye	-0.15	-0.2	-0.06
Right fisheye	0.06	-0.04	-0.28
Back fisheye	-0.02	0.44	-0.37
Long range front	0.02	-0.3	0.07
Long range back left	-0.04	0.31	0.03
Long range back right	0.04	0.49	-0.22

Table B.27: Relative angles in degrees between the poses from our method and the factory calibration for scenario 27.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.63	-0.08	0.13
Front fisheye	-0.2	-0.12	-0.48
Right fisheye	0.1	0.13	-0.21
Back fisheye	-0.14	0.29	-0.6
Long range front	0.17	-0.14	-0.2
Long range back left	-0.14	0.13	-0.76
Long range back right	0.1	0.42	0.03

Table B.28: Relative angles in degrees between the poses from our method and the factory calibration for scenario 28.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.77	-0.22	1.83
Front fisheye	-0.19	0.07	1.43
Right fisheye	0.07	-0.1	1.04
Back fisheye	-0.3	-0.08	1.12
Long range front	-0.13	-0.15	1.37
Long range back left	-0.64	0.16	0.31
Long range back right	-0.43	-0.04	0.98

Table B.29: Relative angles in degrees between the poses from our method and the factory calibration for scenario 29.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.23	-0.24	0.85
Front fisheye	0.08	0.25	0.1
Right fisheye	0.27	0.15	0.3
Back fisheye	-0.01	0.12	0.05
Long range front	0.37	-0.0	0.28
Long range back left	-0.15	0.15	0.09
Long range back right	0.09	0.48	0.3

Table B.30: Relative angles in degrees between the poses from our method and the factory calibration for scenario 30.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.69	-0.4	0.75
Front fisheye	-0.33	-0.24	0.43
Right fisheye	-0.19	-0.07	0.56
Back fisheye	-0.42	0.08	0.46
Long range front	-0.12	-0.17	0.44
Long range back left	-0.39	-0.1	-0.01
Long range back right	-0.24	0.33	0.94

Table B.31: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 31.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.37	0.0	0.09
Front fisheye	-0.3	0.38	-0.08
Right fisheye	-0.13	0.28	-0.18
Back fisheye	-0.23	0.39	-0.04
Long range front	-0.39	0.21	0.03
Long range back left	-0.27	0.22	0.0
Long range back right	-0.39	0.21	-0.05

Table B.32: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 32.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.98	1.21	0.0
Front fisheye	-0.73	1.17	-0.08
Right fisheye	-0.59	-1.67	0.16
Back fisheye	-2.02	0.21	1.78
Long range front	-1.02	0.69	-0.24
Long range back left	0.33	1.06	-0.09
Long range back right	-1.52	-0.76	1.37

Table B.33: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 33.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.53	0.16	0.28
Front fisheye	-0.79	0.19	-0.7
Right fisheye	0.59	-0.62	0.78
Back fisheye	-0.41	0.31	-0.19
Long range front	-0.55	-0.0	-0.08
Long range back left	-0.41	0.13	0.75
Long range back right	-0.46	0.12	-0.71

Table B.34: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 34.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.3	0.24	-0.21
Front fisheye	-0.37	0.64	0.19
Right fisheye	0.4	0.1	0.35
Back fisheye	-0.3	0.47	0.54
Long range front	-0.32	0.49	0.27
Long range back left	-0.05	0.11	0.48
Long range back right	0.07	0.34	0.23

Table B.35: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 35.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.06	0.22	-0.31
Front fisheye	-0.18	0.37	-0.75
Right fisheye	0.04	0.02	-0.65
Back fisheye	0.03	0.52	-0.22
Long range front	-0.15	-0.01	-0.49
Long range back left	-0.07	0.26	0.03
Long range back right	-0.17	0.14	-0.95

Table B.36: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 36.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.05	0.11	-0.07
Front fisheye	0.41	0.49	0.25
Right fisheye	0.54	0.23	-0.03
Back fisheye	0.28	0.14	0.42
Long range front	0.19	0.29	-0.04
Long range back left	0.26	0.31	0.56
Long range back right	0.66	0.36	-0.09

Table B.37: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 37.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	0.21	-0.41	-0.54
Front fisheye	0.29	-0.36	-0.19
Right fisheye	0.3	-0.01	-0.09
Back fisheye	0.22	0.03	0.39
Long range front	0.34	0.13	-0.14
Long range back left	1.39	0.36	1.1
Long range back right	0.06	0.09	0.18

Table B.38: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 38.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.68	-0.38	-0.95
Front fisheye	-0.11	-0.3	-0.94
Right fisheye	-0.07	-0.11	-1.16
Back fisheye	-0.41	0.17	0.03
Long range front	-0.22	-0.27	-0.77
Long range back left	-0.32	-0.18	-0.6
Long range back right	-0.69	0.13	1.36

Table B.39: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 39.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	1.06	0.24	-0.99
Front fisheye	0.59	-0.14	-0.91
Right fisheye	0.54	-0.08	-0.69
Back fisheye	0.81	0.48	0.07
Long range front	0.26	-0.3	-0.92
Long range back left	1.02	0.39	-0.43
Long range back right	0.44	0.11	0.24

Table B.40: Relative angles in degrees between the poses from our method and the dynamic pattern calibration for scenario 40.

Camera	Angle in x-direction	Angle in y-direction	Angle in z-direction
Left fisheye	-0.13	0.01	-0.22
Front fisheye	-0.22	0.07	-0.02
Right fisheye	0.44	-0.24	0.41
Back fisheye	-0.11	0.34	0.31
Long range front	-0.19	0.1	0.14
Long range back left	-0.13	0.27	-0.94
Long range back right	-0.02	0.11	0.0

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY