Towards the Intrinsic Self-Calibration of a Vehicle-Mounted Omni-Directional Radially Symmetric Camera

Sebastian Houben[1]

Abstract— **Intrinsic calibration, *i.e.* finding the mapping between a camera's image positions and corresponding view rays, is a cumbersome, yet unavoidable task in order to accurately generate and interpret results from many kinds of image processing algorithms.**

We address this problem in the context of vehicle-mounted cameras with arbitrary fields of view with applications in advanced driver assistance systems. In particular, we present algorithms to gather the necessary data from unknown scenes and to subsequently estimate the camera parameters. These do rely on vehicle odometry only to resolve the focal scale ambiguity and to recognize when a purely translational motion is performed. We pay special attention to noise handling and circumvention of numerical instabilities.

The proposed pipeline is tested by means of simulations to examine its noise sensitivity. Additionally we calibrate a fisheye camera from a natural scene of only 14 seconds length.

First results show that the self-calibration in natural scenes is eligible and outperforms the straightforward approach of using all calibration parameters from an identically constructed camera.

I. INTRODUCTION

In order to provide comfort and security for both passengers and nearby road users modern vehicles have access to a growing number of more and more sophisticated sensors. Rather than only monitoring the environment and raising a warning in critical situations they also trigger and control braking and evasive manoeuvres. In the future it is to be expected that more and more driving situations will be performed fully autonomously.

An important family of these sensors are video cameras of different building classes. Today these are used for recognition of pedestrians, traffic signs, and other vehicles, as well as localization of arbitrary obstacles, *e.g.* with the help of stereo vision. In order to make use of these results it is imperative to map positions in the camera's image frame to the direction of the respective object w. r. t. the vehicle coordinate system. Determining this mapping is known as calibration.

One distinguishes between intrinsic and extrinsic calibration, the mapping of a frame's image point to its view ray and the mapping of the camera's to the vehicle's coordinate system. To clarify: We define intrinsic calibration as the estimation of all parameters needed to describe the mapping between a world point (w.r.t. the camera position) and an image point. This encompasses focal length, pixel sizes and lens or mirror distortion parameters.[1] In this paper, we focus on the intrinsic calibration and assume that the extrinsic calibration is known to a certain accuracy. This is reasonable since the vehicle chassis defines the position and alignment of the camera. In contrast, even slight misalignments during the camera's and in particular the lens' manufacturing can have a large effect on the intrinsic parameters which makes a calibration unavoidable (*cf*. Fig. 1). In practice, the calibration procedure involves presenting a pattern of known dimensions to the camera from different perspectives, marking known points of this pattern in the camera image, and using the correspondences to compute the lens distortion parameters. We will refer to this procedure as *classical calibration*. Although it has been automated [**?**], this method can be considered cumbersome. Additionally, it is not always possible to perform a classical calibration after, e. g. , a camera has been exchanged in the vehicle.

We present a self-calibration algorithm, a method of estimating a camera's intrinsics without knowledge of the scene. The used camera model is very general and also encompasses wide-angle cameras with a field of view larger than 180°. The proposed algorithm can be divided in three sub-steps: a) the estimation of the radial distortion parameters (Sec. III-B) up to a scalar ambiguity, b) the center of distortion (Sec. III-B), III-C), and c) the determination of the scalar (Sec. III-D).

In order to perform these calculations we rely on odometry data from the vehicle. In the first two sub-steps, this data is only used to ensure that the vehicle is describing a purely – at least approximately – translational motion. In the second step, we utilize the velocity and yaw rate to localize the camera over time w. r. t. a scene-fixed world coordinate system. Furthermore, it is assumed that during self-calibration the change in pitch is limited which can be ensured by an according sensor as well. Although the presence of this odometry readings is very common in modern cars and does not require more than the usual hardware, especially for the third step we rely on an accuracy in motion estimation that is not yet common in today's vehicles. For a discussion on this critical subject, we refer to section V.

We prove the feasibility of our approach by testing it in simulations with different levels of sensor noise. Likewise we calibrate a camera using a short sequence from a driving scenario with moderate speed and compare the results to a traditional calibration (Sec. IV).

[1]Sebastian Houben is with the Institute for Neural Computation, University of Bochum, Germany sebastian.houben(at)ini.rub.de

[1]The notion intrinsic is sometimes defined more narrow in the literature, excluding distortion parameters. For simpler presentation, we deviate from this definition in this paper.

Based on the experimental findings our outline of a self-calibrating vehicle-mounted camera system would be to initialize a newly installed camera with a default intrinsic calibration which will hopefully allow the image processing algorithms to achieve a preliminary sufficient accuracy. In this pre-calibration state the uncertainty of the camera readings should be accordingly reweighed.

At the same time the camera is set to a calibration mode which triggers it to track arbitrary points in the image whilst storing the vehicle odometry (refined by the other calibrated sensors) at the same time. Although much of this data can be discarded immediately the remaining part will allow to perform the calculations presented in this paper. After the residual error has reached a sufficiently low value the camera is then ready to be used for the designed purpose.

II. RELATED WORK

The literature on self-calibration can be divided into two main approaches: calibration under arbitrary and restricted motion. For a general overview over calibration methods we refer to the survey by Puig *et al*. [**?**].

Civera et al. [**?**] showed that the intrinsic parameters of a radially distorted camera [**?**] can be estimated along with the 3d-position and camera location (SLAM) in a combined filtering method, although they rely on a sophisticated Sum-Of-Gaussians filter to handle the nonlinearities. Likewise, Micusík *et al*. [**?**] present a method for estimating the fundamental matrix (*cf*. [**?**]) with the help of a linearized version of the radial distortion function based on the work of Fitzgibbon [**?**].

If the camera motion is controlled or known within a certain accuracy one can arrive at algorithms which usually tend to be numerically more stable. We cite the work by Kelly et al. [**?**] for fusing a visual and an inertial sensor and refer to Ramalingam et al. [**?**] for a perspective on calibration under purely translational and rotational motion.

III. SELF-CALIBRATION

A. Distortion model

In order to describe the camera distortion we use the model proposed by Scaramuzza et al. [**?**]: An image point (ξ, ψ) is written as $(u, v) = (\xi - \xi_c, \psi - \psi_c)$ w.r.t. the center of distortion (ξ_c, ψ_c). We define ξ to be the image column, ψ the row w.r.t. the lower left image corner. This will later yield a right-handed camera coordinate system.

The view ray of (u, v) is denoted by

$$\begin{pmatrix} u \\ v \\ f(r) \end{pmatrix} \tag{1}$$

where f is an invertible function of the variable $r = \sqrt{u^2 + v^2}$. Depending on the camera type, f can take various models. We refer to [**?**] for a detailed overview. In the outline of this algorithm and later experiments we assume that f is a polynomial with the coefficients $\kappa_0, ..., \kappa_n$, a versatile choice encompassing dioptric and catadioptric radially symmetric cameras. The extension to other models is possible and straightforward. Thus, in order to calibrate the camera intrinsically we need to determine $\chi_c, \psi_c, \kappa_0, ..., \kappa_n$. [2]

In reverse, mapping a view ray to an image point one has to set up the equation

$$s \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} u \\ v \\ f(r) \end{pmatrix}$$

with a scalar multiple of the view ray at the left-hand side and u, v, r as unknowns. Normalizing by $\frac{1}{\sqrt{x^2+y^2}}$ parametrizes the ray by r:

$$\frac{r}{\sqrt{x^2 + y^2}} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} u \\ v \\ f(r) \end{pmatrix}$$

We, thus, can find the roots of

$$\frac{r}{\sqrt{x^2 + y^2}} z = f(r) \tag{2}$$

and subsequently retrieve u and v. Please note, that the polynomial (2) should only have one real-valued root that can serve as a radius within the image boundaries.

B. Estimating the radial polynomial

In a first step we will derive a method for estimating $\kappa_0, ..., \kappa_n$. We extend the method by Tardif et al. [**?**] who proposed to use straight line structures from the scene to estimate the radial distortion in the camera image. However, even in very defined scenarios like traffic scenes, it is hard to distinguish between a straight and a slightly curved structure in a distorted image. Therefore, we track the curve a static object point follows during a purely translational movement (*cf*. Fig. 4). In a rectified image this motion would describe a straight line.

Assuming w_1, w_2, w_3 are the view rays of three different points of the curve described in the image, we can state that the vectors lie on a plane and are hence linearly dependent. Thus, the determinant of the concatenated column vectors vanishes.

$$\begin{aligned} 0 = det(w_1, w_2, w_3) &= \begin{vmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ f(r_1) & f(r_2) & f(r_3) \end{vmatrix} \\ &= f(r_1) \begin{vmatrix} u_2 & u_3 \\ v_2 & v_3 \end{vmatrix} - f(r_2) \begin{vmatrix} u_1 & u_3 \\ v_1 & v_3 \end{vmatrix} + f(r_3) \begin{vmatrix} u_1 & u_2 \\ v_1 & v_2 \end{vmatrix} \end{aligned} \tag{3}$$

The last conversion is an application of Laplace's rule. Inserting the polynomial for $f(r)$ yields a linear equation in the unknowns $\kappa_0, ..., \kappa_n$. Using other points from this or other tracks enables us to set up an over-determined system of linear equations. A nonvanishing solution which minimizes the residual lies in the subspace given by the singular vector to the smallest singular value of the coefficient matrix.

[2] The model can be extended by a linear mapping between the image point (u, v) and the x- and y-coordinate of the view ray to account for nonquadratic pixels. We assume that $A = I$. A method for estimating A for wide-angle cameras from an arbitrary image is presented in [**?**].

Fig. 1. The problem at hand: In order to *e.g.* transform a camera image (left) into this helpful ground plane view a representation of the image mapping within the used camera is necessary. The shown ground plane views were computed with different intrinsic parameters: retrieved by a classical calibration method ([**?**], middle left), by the self-calibration technique presented in this paper (middle right), and by the intrinsic parameters of an identically constructed camera (right). Although the intrinsics of an identical camera model were used, the right image contains strong distortions. This emphasizes the need for single camera calibration and the convenience of a self-calibration algorithm.

This vector can be efficiently computed by a singular value decomposition.

Because of memory restrictions it is not possible to consider every possible triplet of view rays. It is, thus, helpful to choose those triplets of points that contribute most to the calibration problem. We found the following reweighing heuristic to yield very satisfying results:

In a first step, we choose the first, last, and a central point from a track and set up the system of equations (3) with one equation per track. The choice is reasonable because the respective view rays will span the maximum angle and hence contain information about the distortion of the largest possible image region. This yields a preliminary solution which allows us to assign weights to each view ray triplet.

It is evident that the larger the angle between the first and the last point of a track the more information does its curve in image coordinates contain about the distortion. Hence, we use the preliminary solution from the first step to estimate this angle ϕ and reweigh each linear equation by a factor of $\sqrt{\phi}$.

C. Estimating the center of distortion

The center of distortion (ξ_c, ψ_c) defines an image point as the origin of the radial distortion and, thus, is a critical parameter of the whole calibration. Introducing (ξ_c, ψ_c) to (3) as a free variable results in nonlinearities and a multimodal fitness landscape. We therefore suggest to begin with a coarse grid search around the image center and start a steepest-descent from the best grid point. For every candidate distortion center we set up the system (3), retrieve coefficients for the radial distortion polynomial and compute the residual error by the following means:

First, we compute the view rays $w_1, ..., w_m$ for each track, *e.g.* $(u_1, v_1), ..., (u_m, v_m)$. Due to errors in the estimation of the polynomial coefficients and in the tracking process $w_1, ..., w_m$ will, in general, not be linearly dependent which we demanded in system (3). We thus compute a least-squares normal vector n as the singular vector to the smallest singular value of the system

$$\begin{aligned} w_1^T n &= 0 \\ &\vdots \\ w_m^T n &= 0 \end{aligned} \tag{4}$$

and use n to project the view rays on the common plane. Let us denote those corrected view rays by $w'_1, ..., w'_m$.

$$w'_k = w_k - (w_k^T n)n$$

Depending on the tracking errors and the current distortion center candidate, (4) can be ill-posed. We detect this by examining the ratio between the smallest and the second smallest singular value. If this ratio becomes too large, this is a strong indication that the distortion center candidate is infeasible and we mark the track as *not reconstructible*.

Second, we reproject the corrected view rays $w'_1, ...w'_m$ to the corrected image points $(u'_1, v'_1), ..., (u'_m, v'_m)$ and compute the residual error as

$$\epsilon = \sum_{i=1}^{m} ||(u'_i, v'_i) - (u_i, v_i)||^2 + \alpha\eta_p + \beta\eta_t \tag{5}$$

As stated in section III-A the reprojection requires the roots of a polynomial. If more than one root is real and lies within the image coordinate range, we mark that point *not reprojectable*.

The geometric error is incremented by values α, β multiplied by the number of nonreprojectable tracks η_t and nonreprojetable points η_p. It is reasonable to choose α, β high enough to dominate the sum (5) in order to prefer candidates with the least number of infeasible tracks and points.

D. Eliminating the scalar ambiguity

Every scalar multiple of the solution of (3) is a solution itself. This is intuitively clear because we did not imply any information of the observed scene's size[3]. In order to estimate the unknown scalar we assume that we now have framewise (potentially noisy) position and orientation information of the camera that we can obtain from the vehicle's odometry data. This information is denoted by the translation vector t_j and the 3×3 rotation matrix R_j at frame j. Thus

$$R_j p - R_j t_j$$

will transform the scene point p to the camera coordinate system.

[3]However, this intuition is not completely accurate. We have to handle a focal ambiguity, not a scale ambiguity.

Let us assume that we have tracked a point in two frames k and l and computed the preliminary view rays

$$w_k = \begin{pmatrix} u_k \\ v_k \\ sf(r_k) \end{pmatrix} = \underbrace{\begin{pmatrix} u_k \\ v_k \\ 0 \end{pmatrix}}_{=:w_k^{uv}} + s \underbrace{\begin{pmatrix} 0 \\ 0 \\ f(r_k) \end{pmatrix}}_{=:w_k^{f}}$$

$$w_l = \begin{pmatrix} u_l \\ v_l \\ sf(r_l) \end{pmatrix} = \underbrace{\begin{pmatrix} u_l \\ v_l \\ 0 \end{pmatrix}}_{=:w_l^{uv}} + s \underbrace{\begin{pmatrix} 0 \\ 0 \\ f(r_l) \end{pmatrix}}_{=:w_l^{f}}$$

via our model from section III-B. Our goal is to compute s.

Since the view rays intersect in the world coordinate system, we demand them and the line between the two camera centers to lie on a plane. Again, the determinant of the respective concatenated vectors is zero:

$$\begin{aligned} & 0 \\ =& det(R_k^T w_k, R_l^T w_l, t_k - t_l) \\ =& \underbrace{det(R_k^T)}_{=1} \cdot det(R_k R_k^T w_k, R_k R_l^T w_l, R_k(t_k - t_l)) \\ =& det(w_k, R_k R_l^T w_l, R_k(t_k - t_l)) \\ =& det(w_k^{uv} + s w_k^f, R_k R_l^T (w_l^{uv} + s w_l^f), R_k(t_k - t_l)) \\ =& det(w_k^{uv}, R_k R_l^T w_l^{uv}, R_k(t_k - t_l)) \\ & + s \cdot det(w_k^f, R_k R_l^T w_l^{uv}, R_k(t_k - t_l)) \\ & + s \cdot det(w_k^{uv}, R_k R_l^T w_l^f, R_k(t_k - t_l)) \\ & + s^2 \cdot det(w_k^f, R_k R_l^T w_l^f, R_k(t_k - t_l)) \end{aligned} \tag{6}$$

The transformations use the properties of the determinant and the orthogonality of rotation matrices. They result in a polynomial of degree 2 which can be solved for s. In this ansatz we avoided implying distance relations which avoids strong nonlinearities.

There are a few special cases that deserve investigation: If the camera movement is purely translational ($R_k = R_l$), the coefficient of s^2 will vanish because the upper left 2×2 sub-determinant is 0. Thus, the equation becomes linear which simplifies matters. However, if the translation is additionally perpendicular to the camera orientation, $R_k(t_k - t_l)$ will vanish in the z component and the constant term will drop leaving $s = 0$ as the only, obviously infeasible, solution of the equation. Likewise, if the translation is directed along the optical axis, (u_k, v_k) will be a multitude of (u_l, v_l) and thus all determinants go to 0 making every s a solution. In conclusion, depending on the specific geometric relation of the cameras numerical instabilities will occur.

Similar to the system (3) we set up (6) for chosen pairs of view rays from a scene point with a known pair of corresponding camera locations. To circumvent the aforementioned instabilities we implement a RANSAC-like method where we randomly choose a single equation and compute the residual error for all other equations several times. We form a consensus set of those equations that have a residual error less than a given threshold and choose the largest such consensus set to solve for s.

IV. EXPERIMENTS

We evaluate our approach by means of simulations and images from a scene recorded by a vehicle-mounted fisheye camera.

A. Simulations

Using a distortion model that we received with classical calibration approaches presented in [**?**], we create tracks of varying length, orientation, and measurement noise. As a baseline method, we suggest to randomly choose triplets from each track to set up a system of 1000 equations from the type (3). This is compared to the proposed method from section III-B where we apply the heuristic for choosing and reweighing the view ray triplets. Fig. 2 shows the performance of both methods for varying point position noise and length of tracks.

Second, we test the sensitivity of the focal scale estimator (*cf*. Sec. III-D) with respect to inaccuracy in camera position and orientation. The experiment is performed with a short simulated traffic scenario where the vehicle describes a narrow curve (yaw rate 3° / frame) while the sensors track bypassing points on the ground plane. The sequence consists of 100 frames with 37 tracks. Fig. 3 shows the goodness of the focal scale estimation with respect to growing camera location uncertainty.

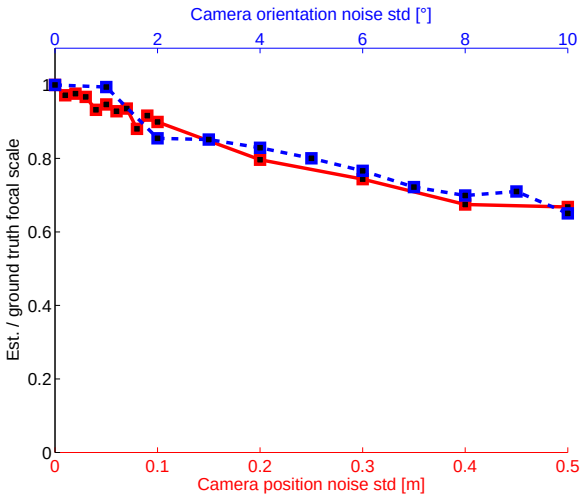

Fig. 3. The sensitivity of the focal scale estimation method (*cf*. Sec. III-D) for varying translation (red solid) and rotation angle noise (blue dashed). The performance is quantified by the ratio of the ground truth over the estimated camera model's focal scalar.

B. Real-world data

As a first test for our algorithm to work with real-world data we choose a sequence recorded at a parking lot where the ego-vehicle steers into and subsequently passes a row of parked cars at moderate speed. The scenario is no longer than 210 frames (14 s). It contains a curve motion of about 80 frames followed by a purely translational motion of 130

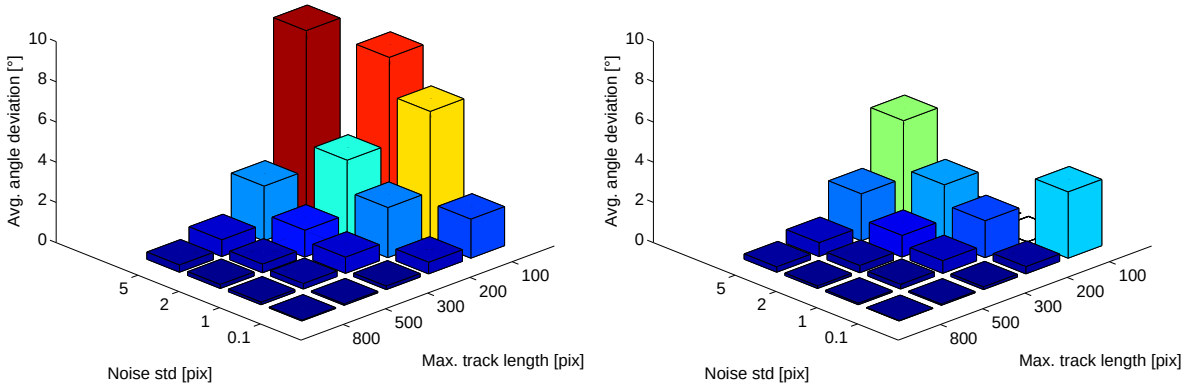

Fig. 2. The figure shows a comparison of the baseline method and the method proposed in section III-B. The goodness of an estimated distortion model is measured by the average angular deviation between the estimated and the correct view rays for each image pixel. Every calibration trial was executed with 100 input tracks.

frames. The camera is a fisheye camera mounted at the right side mirror (height: 91cm) with a maximum view angle of 192. The resolution of the acquired camera image was 1280 $\times$ 800 pixels at 15 frames per second.

The image processing pipeline consists of a *Harris corner detector* to identify points of interest and a multi-scale template matching to framewisely track those points. In order to handle the strong distortion, the matching is also performed with horizontally or vertically squeezed versions of the point template. Altogether every point is represented by three differently scaled templates each squeezed to 3×3 aspect ratios. We found this effort necessary to track features over a sufficient distance. In order to track a point each of its squeezed templates is searched in a surrounding via normalized cross-correlation (NCC) and the one with maximum NCC coefficient determines the match. The tracking can thus deal with distortions without relying on a precalibration. As an alternative approach we manually labeled a number of points over the sequence yielding an outlier-free, yet not subpixel-accurate input data set. Refer to Fig. 4 for a comparison. The egomotion was retrieved by labeling landmarks in other vehicle-mounted cameras that we calibrated in the classical fashion beforehand. The motion estimation was done framewisely yielding cumulative errors (drift) as customary odometry sensors do. [4]

We executed the entire calibration pipeline on the input tracks using those from the translational motion for distortion estimation (*cf*. Sec. III-B, Sec. III-C, 91 automatically and 21 manually tracked points) and the rotational motion for focal scale estimation (*cf*. Sec. III-D, 151 automatically and 15 manually tracked points).

Fig. 5 shows the course of the estimated radial distortion function $f(r)$ (*cf*. Sec. III-A). The center of distortion deviated by 9.44 and 2.55 pixels for the automatically, and by 10.30 and 12.78 for the manually tracked points, in column and row direction respectively. Furthermore, the smaller errors of the results retrieved by the automatically tracked points show that their large number stabilizes the output and circumvents the problem of outliers.

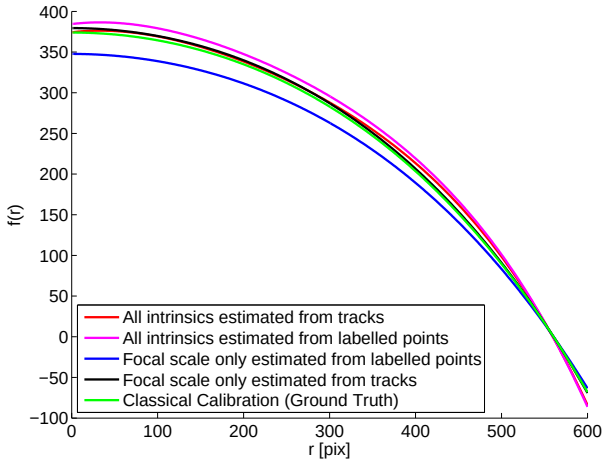

Fig. 5. The estimation of the radial distortion function $f(r)$ retrieved by automatically and manually tracked scene points. For the graphs labeled "Focal scale only" the polynomial coefficients $\kappa_0, ..., \kappa_n$ were taken from the classical calibration (ground truth).

V. DISCUSSION

The simulations showed that the whole self-calibration pipeline is eligible to estimate useful camera intrinsics under presence of moderate noise. Our extension of Tardif's method [**?**] (*cf*. Sec. III-B) clearly outperforms the baseline approach considering noise sensitivity and reliance on larger tracks (*cf*. Fig. 2). Additionally, the method proposed in section III-D allows to resolve the focal scale ambiguity at a light camera location uncertainty of 5 cm and an orientation uncertainty of 1 to 2 degrees. Our analysis shows that this step is indeed critical in the presence of noise.

This level of accuracy in odometry readings can surely not be achieved with customary hardware deployed in today's vehicles. However, we would like to point out that for the

[4]Unfortunately in the sequences we had at our disposal the vehicle odometry had not been recorded, thus, forcing us to follow this workaround.

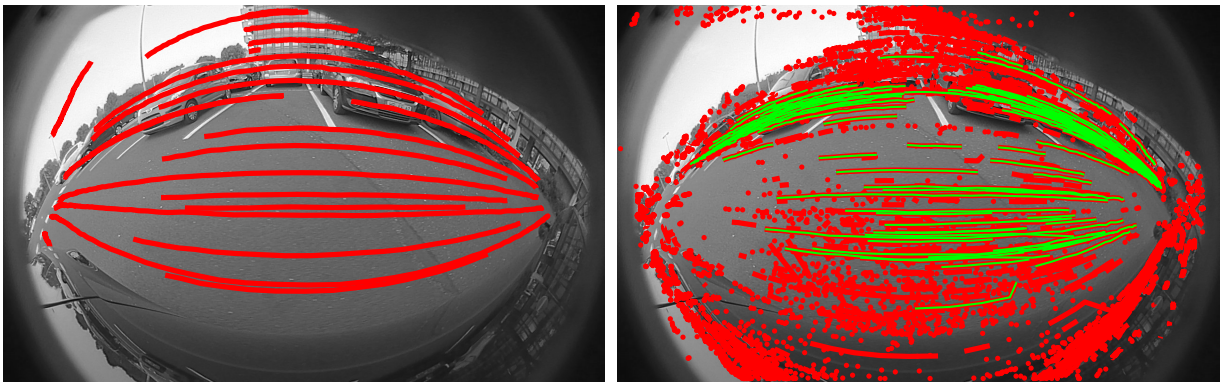

Fig. 4. The tracked points for self-calibration were gathered manually (left) and by a multi-scale template matching approach (right). The automatically acquired tracks were filtered for length and smoothness. The green curves indicate those which were passed on to the calibration stage.

main part of the algorithm only a straight translational motion is required which can reasonably be verified using only the steering wheel angle, a rather precise reading. Furthermore it should be kept in mind that sensor precision will grow as SLAM and egomotion estimation methods (on other sensors than the cameras that we intend to calibrate) will become widespread in modern vehicles. Thus, we claim that after replacement of an omnidirectional camera, the readings from the remaining sensors will be precise enough to guarantee for a useful self-calibration.

The results on the real-world traffic sequence support the results from the simulations. It is indeed possible to reasonably calibrate the wide-angle camera by tracking scene points. The results also surpass the ones retrieved by use of the manually labeled points (*cf*. Fig. 5) although this can be accounted to the relatively small number of tracks. We should point out that the tracking algorithm was not real-time capable. It is obvious that a sufficiently small number of tracks followed at the same time will amend this problem, albeit at the cost of longer sequences in order to find sufficiently many stable points. The estimation algorithm itself was implemented in Matlab and took less than three seconds to execute on a standard desktop computer.

Finally, comparison with the parameters of an identically constructed camera reveals that the self-calibration is in fact a better solution than initializing the camera with a default calibration.

VI. CONCLUSIONS

In future work we want to look more deeply into the handling of outliers which were strongly reduced within our tests compared to natural traffic scenes. These outliers can be caused by nonstatic scene points, *e.g.*, from other vehicles, or reflecting surfaces. We furthermore will examine the performance of our approach on larger datasets with faster driving speed, different camera models and mounting positions.