IEEE copyright notice

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redis- tribution to servers or lists, or to reuse any copyrighted com- ponent of this work in other works must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O. Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 908-562-3966.

Automatic Extrinsic Camera Self-Calibration Based on Homography and Epipolar Geometry

Michael Miksch and Bin Yang
Chair of System Theory and Signal Processing
University of Stuttgart, Germany
{michael.miksch,
bin.yang}@lss.uni-stuttgart.de

Klaus Zimmermann
European Technology Center (EuTEC)
Sony Deutschland GmbH
D-70327 Stuttgart, Germany
klaus.zimmermann@sony.de

***Abstract*— In this paper we present a method to calibrate the extrinsic parameters of a monocular camera on a moving vehicle. The method is based on a homography between two camera shots. Therefore, only the road surface has to be visible in the pair of images. A reasonable definition of the vehicle coordinate system in combination with the use of epipolar geometry reduces the complexity to parameterize the underlying homography matrix. The extrinsic parameters are determined analytically by two correctly matched feature points located on the road surface. The final parameter set is determined by a recursive filter which considers various estimates over time. Results with a real-world video sequence indicate that the method is comparable to classical offline calibration techniques using objects of known geometry.**

I. Introduction

Camera calibration is an important area in computer vision. It establishes the relationship between the 3D environment and its projection onto the image plane. The task of calibration can be subdivided into two parts: the intrinsic and the extrinsic calibration. The extrinsic parameters describe the relative position and orientation of the camera with respect to the *vehicle coordinate system* (VCS). The intrinsic parameters model the projection of points from the *camera coordinate system* (CCS) onto the image plane.

The intrinsic and extrinsic parameters are often estimated offline before the initial operation of the system. Methods like [1]–[4] make use of calibration objects of known geometry. As such objects are not available during the runtime of the system, the parameters are then kept constant. The assumption that the intrinsic parameters are static is reasonable for automotive applications. In contrast, mechanical stress, like when the vehicle hits a bump, can cause a drift of the extrinsic parameters which degrades the performance of the complete system. There is, therefore, a demand to automatically recalibrate the extrinsic parameters online because a repetitive calibration with offline methods is not practical for automotive applications. This paper provides an efficient solution to this problem.

Camera self-calibration refers to methods which do not require calibration objects. They continuously estimate the parameters during the runtime of the system. Assumptions on the structure of the road are commonly used as an alternative. The paper [5] assumes road boundaries to be straight and flat. Similarly, the work [6] assumes straight lane markings. Dashed and periodic lane markings are used for calibration in [7]. Some of the previous principles additionally assume that either the vehicle moves in a straight line, or the camera height or velocity information is available, or a combination of all. These assumptions are often flawed under real-world automotive conditions and lead to inaccurate extrinsic parameters. At least for a side view camera, the previous methods are not applicable to solve the calibration problem.

In this work, only the road surface has to be visible in the images and is assumed to be approximately flat in the immediate vicinity of the camera. The VCS is defined with respect to this plane and the vehicle. This enables the estimation of the extrinsic parameters, since they describe the relationship between the CCS and the VCS. The image motion on the road surface, which is induced by the movement of the camera, can be described by a projective transformation, also known as homography. The extrinsic parameters are part of this homography and can be determined if the homography is estimated from the image motion.

Our previous work [8] relies on odometric data and minimizes a one-dimensional cost function to estimate the homography matrix. The classical estimation of the homography suffers from the fact that a problem of eight *degrees of freedom* (DOF) has to be solved. At least four feature pairs on the road plane are required to find a unique solution to this problem. A feature pair is the assignment of two image points which are projections of the same object point in 3D. Standard methods tend to fail to extract and assign four feature pairs on the road surface. We propose a new approach for the automatic calibration of the extrinsic parameters. We first estimate the essential matrix of the epipolar geometry from feature pairs of successive images. This is much easier than the estimation of the homography since the feature pairs can be located anywhere, not only on the road surface. The epipolar geometry is then used to identify feature points on the road surface and to simplify the parameterization of the homography matrix. Finally, only two feature pairs on the road surface are required to analytically determine the extrinsic parameters.

The outline of this paper is as follows: In Sec. II we introduce the fundamentals. The proposed camera calibration method is described in Sec. III. The results of the method for a real-world video sequence are presented in Sec. IV. Finally, a conclusion is given in Sec. V.

II. Basics

A. Camera projection

The underlying camera model is the pinhole camera. It describes the projection of a 3D object point $\mathbf{M}_v = [X, Y, Z]^T$ onto its image point $\mathbf{m}_p = [u_p, v_p]^T$ by

$$\lambda\,[\mathbf{m}_p^T, 1]^T = \mathbf{A}\,[\mathbf{R}_e\ \mathbf{t}_e]\,[\mathbf{M}_v^T, 1]^T \tag{1}$$

where λ is a scalar factor for the normalization, see (4), $[\mathbf{R}_e\ \mathbf{t}_e]$ are the extrinsic parameters, and $\mathbf{A}$ is the intrinsic matrix. The intermediate step

$$\lambda\,[\mathbf{m}_c^T, 1]^T = \mathbf{M}_c = \mathbf{R}_e\,\mathbf{M}_v + \mathbf{t}_e \tag{2}$$

transforms the point $\mathbf{M}_v$ from the VCS to the point $\mathbf{M}_c$ in the CCS by an Euclidean transform. $\mathbf{R}_e$ is a rotation matrix with $\mathbf{R}_e^{-1} = \mathbf{R}_e^T$, and $\mathbf{t}_e$ is a translation vector, resulting in normalized coordinates $\mathbf{m}_c = [u_c, v_c]^T$. The intrinsic transform between the normalized and the image coordinates is defined by

$$\widetilde{\mathbf{m}}_p = \mathbf{A}\,\widetilde{\mathbf{m}}_c \quad \text{and} \quad \widetilde{\mathbf{m}}_c = \mathbf{A}^{-1}\,\widetilde{\mathbf{m}}_p. \tag{3}$$

Here, ”$\sim$” represents homogeneous coordinates. The relationship between Cartesian coordinates $\mathbf{m}$ and homogeneous coordinates $\widetilde{\mathbf{m}}$ is

$$\lambda \in \Re\backslash\{0\} : \lambda\,[\mathbf{m}^T, 1]^T = \widetilde{\mathbf{m}}. \tag{4}$$

Therefore, homogeneous coordinates can be scaled arbitrarily while maintaining the representation of the same point.

B. Vehicle coordinate system (VCS)

Let us consider a VCS whose origin is located on the road surface and below the camera's *center of projection* (COP). The z-axis of the VCS is pointing vertically towards the camera, whereas the x-axis is pointing parallel to the lateral profile of the vehicle into the direction of travel.

C. Extrinsic parameters

The extrinsic parameters $[\mathbf{R}_e\ \mathbf{t}_e]$ are already defined in (2). The Euclidean transform can be inverted as follows

$$\mathbf{M}_v = \mathbf{R}_e^T\,\mathbf{M}_c + \mathbf{t}_h \quad \text{with} \quad \mathbf{t}_h = -\mathbf{R}_e^T\,\mathbf{t}_e. \tag{5}$$

Based on the assumption in II-B, the COP has the coordinates $\mathbf{M}_c = 0$ in the CCS and $\mathbf{M}_v = \mathbf{t}_h = [0, 0, h_c]^T$ in the VCS where h_c denotes the camera height. Since

$$\mathbf{t}_e = -\mathbf{R}_e\,\mathbf{t}_h, \tag{6}$$

the extrinsic parameters can alternatively be expressed by the rotation matrix $\mathbf{R}_e$ and the camera height h_c.

D. Motion of the vehicle

The motion of the vehicle between two camera shots is described by an Euclidean transform in the VCS

$$\mathbf{M}_v^{'} = \mathbf{R}_v\,\mathbf{M}_v + \mathbf{t}_v. \tag{7}$$

In the CCS, the same motion is described by $\mathbf{M}_c^{'} = \mathbf{R}_c\,\mathbf{M}_c + \mathbf{t}_c$ where $\mathbf{R}_c$ and $\mathbf{t}_c$ are derived in Appendix I. The relationship between different coordinates of the same object point in VCS and CCS before and after the motion is illustrated in Fig. 1. The extrinsic parameters are assumed to be identical for both camera shots.

$$\begin{tikzcd}
 & \text{image 1} & \text{image 2} \\
\text{VCS} & \mathbf{M}_v \arrow[r, "{[\mathbf{R}_v\ \mathbf{t}_v]}"] \arrow[d, "{[\mathbf{R}_e\ \mathbf{t}_e]}"] & \mathbf{M}_v^{'} \arrow[d, "{[\mathbf{R}_e\ \mathbf{t}_e]}"] \\
\text{CCS} & \mathbf{M}_c \arrow[r, "{[\mathbf{R}_c\ \mathbf{t}_c]}"] & \mathbf{M}_c^{'}
\end{tikzcd}$$

Fig. 1. Relationship between different coordinates

E. Epipolar geometry

The image motion between two camera shots depends on the distance of the considered object point to the COP, and on the motion of the camera. Nevertheless, the corresponding point in the second image is located on a straight line, the so-called epipolar line, for a fixed point in the first image and vice versa. This relationship is part of the epipolar geometry (see [9]) and is formulated in normalized coordinates as

$$\mathbf{E} = [\mathbf{t}_c]_\times\,\mathbf{R}_c. \tag{8}$$

$\mathbf{E}$ is called the essential matrix and $[.]_\times$ denotes the skew-symmetric matrix operator

$$[\mathbf{t}_c]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad \text{with} \quad \mathbf{t}_c = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{9}$$

The equivalent relationship in image coordinates is described by the fundamental matrix $\mathbf{F} = \mathbf{A}^{-T}\,\mathbf{E}\,\mathbf{A}^{-1}$. The epipolar line is computed as follows

$$\mathbf{l} = \mathbf{F}\,\widetilde{\mathbf{m}}_p \tag{10}$$

Consequently, the corresponding image point fulfills the equation

$$\widetilde{\mathbf{m}}_p^{'T}\,\mathbf{F}\,\widetilde{\mathbf{m}}_p = 0, \tag{11}$$

the so-called epipolar constraint. The constraint can also be expressed in normalized coordinates as follows

$$\widetilde{\mathbf{m}}_c^{'T}\,\mathbf{E}\,\widetilde{\mathbf{m}}_c = 0. \tag{12}$$

Unfortunately, points which fulfill the epipolar constraint do not necessarily belong to the same object point.

F. Homography of the road surface

In contrast to Sec. II-E, points on a plane have a certain distance to the COP and the image motion is no longer ambiguous. This leads to a homography between two camera shots which can be expressed by the matrix

$$\mathbf{H}_c = \mathbf{R}_e \left(\mathbf{R}_v + \frac{\mathbf{t}_v\,\mathbf{n}_v^T}{-h_c} \right) \mathbf{R}_e^T \tag{13}$$

where $\mathbf{n}_v = [0, 0, 1]^T$ is the normal vector of the road surface (see Appendix I for a detailed derivation). Note that the transformation is given in normalized coordinates. The motion of a point on the road surface is defined by

$$\widetilde{\mathbf{m}}_p^{'} = \mathbf{A}\,\mathbf{H}_c\,\mathbf{A}^{-1}\,\widetilde{\mathbf{m}}_p \quad \text{and} \quad \widetilde{\mathbf{m}}_c^{'} = \mathbf{H}_c\,\widetilde{\mathbf{m}}_c \tag{14}$$

in image and normalized coordinates, respectively.

III. Calibration

A. Assumptions

First of all, we will summarize the assumptions used by our calibration method:

- The road is flat in the vicinity of the origin of the VCS.
- The intrinsic parameters are available and remain constant (intrinsic matrix $\mathbf{A}$ is known).
- The camera height h_c is known, but is not needed for the calibration of $\mathbf{R}_e$.
- The vehicle is traveling in a straight line between two consecutive frames, i.e. $\mathbf{R}_v = \mathbf{I}$ and $\mathbf{t}_v = [-\Delta s, 0, 0]^T$. Δs denotes the distance which the camera has moved into the direction of travel.
- There is no odometric data (e.g. yaw rate or velocity sensor) available from the vehicle (Δs unknown).

The fourth assumption is motivated by a detailed study of the motion of a vehicle. The statistical analysis of more than $20,000$ km measured data reveals that the yaw rate is less than $0.65°/s$ for 50% of the time, less than $1.07°/s$ for 75% of the time, less than $1.65°/s$ for 90% of the time and greater than $2.19°/s$ for only 5% of the time. A camera in an automotive application normally operates with frame rates in the range of $10 - 50$ fps. Therefore, the rotation of the camera between two consecutive frames is negligible most of the time.

B. Calibration principle

The extrinsic calibration aims at identifying the extrinsic parameters $[\mathbf{R}_e\ \mathbf{t}_e]$. The task is, therefore, to find a homography matrix, which best fits the real image motion on the road surface between two consecutive frames. In Fig. 2, two successive images of a video sequence are shown. The real image motion is depicted by the feature pair indicated either by the green line, or the red rectangular region in the top image and its transformed version in the frame below.

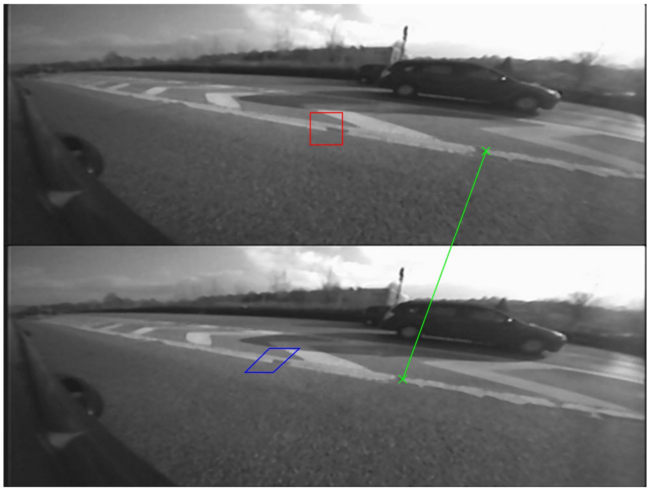

Fig. 2. Two consecutive frames of a video sequence with one exemplary feature pair (green line) and a rectangular region of size 32x32 (red) in the top image and the equivalent transformed region (blue) in the bottom image

There is a unique homography matrix $\mathbf{H}_c$ which exactly describes this displacement for all feature pairs extracted from the road surface (see II-F). On the other hand, if there is a set of feature pairs available on the road, a homography matrix can be estimated, from which the extrinsic parameters can be extracted. This is our basic concept for the calibration of the extrinsic parameters.

A general projective transformation (homography matrix) has 8 DOF. It is known that a minimum of four feature pairs is needed for a unique solution of $\mathbf{H}_c$. The estimation on the basis of four feature points tends to be defective because the correct localization of four points on the road surface is difficult. We will reduce the required number of feature pairs to a minimum of two by taking advantage of epipolar geometry. The major steps of this calibration method are summarized in Fig. 3.

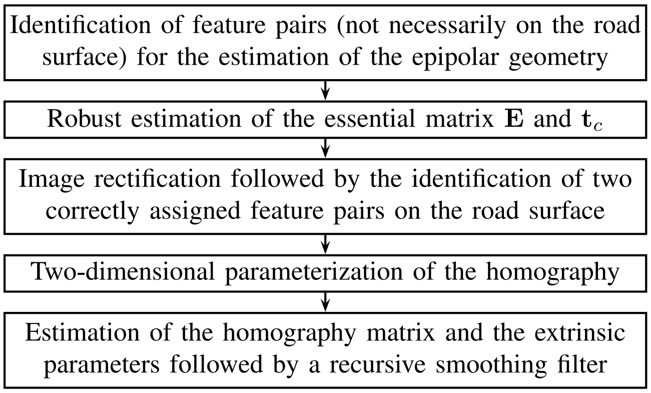
Identification of feature pairs (not necessarily on the road surface) for the estimation of the epipolar geometry
Robust estimation of the essential matrix $\mathbf{E}$ and $\mathbf{t}_c$
Image rectification followed by the identification of two correctly assigned feature pairs on the road surface
Two-dimensional parameterization of the homography
Estimation of the homography matrix and the extrinsic parameters followed by a recursive smoothing filter

Fig. 3. Overview of the major steps of the self-calibration method

C. Estimation of the epipolar geometry

The essential matrix $\mathbf{E}$ is estimated by using feature pairs, but with the advantage that they can be located anywhere, not only on the road surface. Standard methods like the *Scale-Invariant Feature Transform* (SIFT [10]) or the *Speeded Up Robust Features* (SURF [11]) can be applied to extract the feature pairs. These methods are well known in literature for the estimation of the epipolar geometry. We use a *Harris corner detector* in combination with a *sum of absolute difference* (SAD) block matching strategy, since this is much faster and still suitable for our purpose.

For an estimation of the essential matrix in the presence of mismatches of the feature pairs, methods like *RANdom SAmpling Consensus* (RANSAC [12]) or *Least Median of Squares* (LMedS [13]) are proven to be robust. Both methods have in common that, for each iteration, first a set of feature pairs is randomly selected to determine one instance of the essential matrix and secondly, all other feature pairs are tested to see whether they fit that particular matrix. If a feature pair fits the essential matrix, it is marked as a so-called inlier, whereas all other pairs are marked as outliers. The only difference between both methods is the criterion to assign a pair as an inlier or outlier. RANSAC uses a static threshold, whereas LMedS adopts the threshold in such a way that always half of the feature pairs are marked as inliers. We refer to the excellent work of [14], [9] and [15] for further information about the robust estimation of the essential matrix.

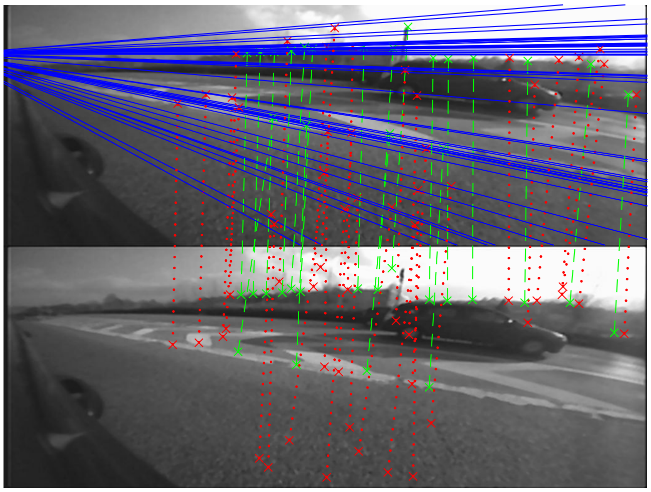

Fig. 4. Two consecutive frames of a video sequence with several feature pairs: identified as inliers (green, dashed) and outliers (red, dotted) and the corresponding epipolar lines (blue, solid)

Fig. 4 shows the feature pairs extracted from two consecutive frames. Inliers are marked green and outliers are marked red. The blue lines represent the corresponding epipolar lines for each feature pair.

As mentioned before, a randomly selected set of feature pairs is used to estimate the essential matrix. Normally, linear methods like the 7-point or 8-point algorithm [16] are applied for that purpose. The names of the algorithms suggest that they need at least seven or eight feature pairs. The feature pairs are used in combination with (12) to form a *system of linear equations* (SLE) which is solved to determine $\mathbf{E}$. The SLE does not take into account that the essential matrix is rank deficient by definition ($\text{rank}(\mathbf{E}) = 2$). Accordingly, the solution of the SLE is corrected afterwards to fulfill that constraint [9]. The determination of the essential matrix on the basis of seven or eight feature pairs is actually overdetermined, since the rotation matrix has 3 DOF and the translation vector has 2 DOF. This is a drawback in the presence of mismatches because the more feature pairs that are needed, the more iterations are required to reliably select at least one uncorrupted set of pairs.

We require only two feature pairs for the intermediate estimation of the essential matrix because no rotation is assumed (see III-A). The matrix $\mathbf{R}_c$ is, therefore, the identity matrix which leads to the new essential matrix $\mathbf{E} = [\mathbf{t}_c]_\times$ according to (8). Each feature pair, which is inserted into (12), now results in the following equation

$$[v_c - v_c^{'}, u_c^{'} - u_c, u_c v_c^{'} - v_c u_c^{'}]\, \mathbf{t}_c^T = 0. \tag{15}$$

Two feature pairs and the additional constraint $\|\mathbf{t}_c\| = 1$ are sufficient to find a unique solution of $\mathbf{t}_c$ and $\mathbf{E}$. The complete estimation process of the essential matrix is not covered by this paper in detail. It is worthwhile to mention that the simplification leads to a faster convergence because less iterations are needed. In the following, it is assumed that the translation vector $\mathbf{t}_c$ is reliably estimated. Note that the real length of the vector can not be determined.

D. Feature extraction on the road plane

The estimation of the extrinsic parameters requires feature pairs on the road surface. In principle, a subset of the feature pairs, which are already used for the estimation of the essential matrix in III-C, could be reused. At this point, a different technique of the extraction and matching is introduced because the standard methods tend to fail for feature points on the road surface. It takes advantage of the epipolar geometry, namely the known translation vector $\mathbf{t}_c$. The requirements are summarized as follows:

- feature pairs should be located on the road plane,
- each pair should fulfill the epipolar geometry,
- sub-pixel accuracy of the position of the feature pairs,
- mismatches should be recognized.

In general, the correspondence problem is a two-dimensional problem, since the two-dimensional coordinates of the corresponding feature point have to be identified. To simplify the feature extraction and matching, we transform the image in a similar manner to the image rectification process in stereo vision does. The original and the transformed version of an image is shown in Fig. 5 as illustration of this transformation.

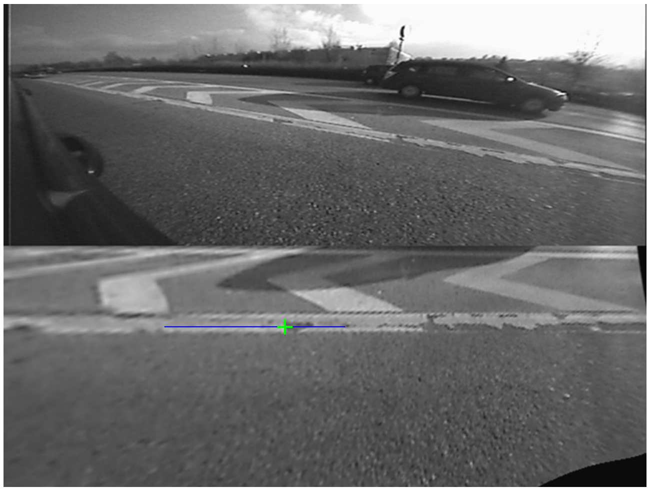

Fig. 5. The original and the transformed version of an image, one exemplary feature point extracted on the road surface (green cross), the horizontally aligned scan line for the disparity estimation (blue line)

The image resampling is based on a projective transformation. Primarily, the underlying transformation matrix $\mathbf{T}$ consists of the original intrinsic matrix $\mathbf{A}$ and a new intrinsic matrix $\mathbf{V}$ (ideal pinhole camera) as follows

$$\mathbf{T}^{-1} = \mathbf{A}\, \mathbf{R}_T\, \mathbf{V}^{-1} \tag{16}$$

with

$$\mathbf{R}_T = \begin{bmatrix} \frac{1}{a_x} \begin{bmatrix} -t_x \\ -t_y \\ -t_z \end{bmatrix}, \frac{1}{a_y} \begin{bmatrix} t_y \\ -t_x \\ 0 \end{bmatrix}, \frac{1}{a_z} \begin{bmatrix} t_x\, t_z \\ t_y\, t_z \\ t_x^2\, t_y^2 \end{bmatrix} \end{bmatrix} \tag{17}$$

where the matrix $\mathbf{R}_T$ rotates the original perspective to the new axially parallel alignment. The parameters a_x, a_y and a_z normalize the column vectors of the rotation matrix. Note that the matrix $\mathbf{R}_T$ is orthogonal and derived from the vector $\mathbf{t}_c$ defined in (9). The rotation is based on the idea that the first column vector of the matrix $\mathbf{R}_T$ rotates the x-axis of the new perspective to the baseline between the two centers of projection represented by $\mathbf{t}_c$, the second column vector is chosen to be perpendicular to this baseline and the original z-axis, and the third column vector is computed as the cross product of the first and second column vector.

The corresponding feature point is located along a one-dimensional scan line after the transformation, which is horizontally aligned. Actually, a scan line represents an epipolar line of the new perspective. The correspondence search would also be feasible along the original epipolar line but the processing based on the transformed image has a couple of advantages. First of all, the matrix $\mathbf{V}$ provides the opportunity to select a *region of interest* (ROI) which contains potential feature points on the road plane. Furthermore, the feature points can be extracted by a simple horizontal gradient filter followed by a non-maximum suppression. One exemplary feature point $\mathbf{m}_t = [u_t, v_t]^T$ is depicted in Fig. 5 as a green cross. The intensity values along the blue line are depicted in Fig. 6, where the high gradients are obviously around the extracted feature point. The red intensity curve is extracted from the subsequent image, which is transformed identically. The displacement of the feature point is indicated by the arrow (≈ 70 pixels).

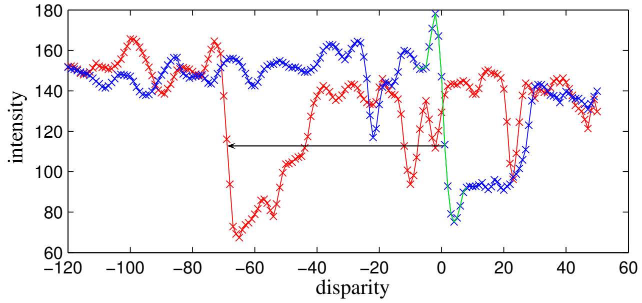

Fig. 6. The underlying data basis for the disparity estimation, the intensity values from the previous (blue) and current (red) transformed image

The displacement Δd_i, also called disparity, can be determined by solving

$$\Delta d_i = \mathrm{argmin}_{d \in [\Delta d_{min}, \Delta d_{max}]} \sum_{i=-b}^{b} |\mathbf{e}(d,i)|^2 \qquad (18)$$

with

$$\mathbf{e}(d,i) = \mathbf{I}_1(u_t + i, v_t) - \mathbf{I}_2(u_t + i + d, v_t) \qquad (19)$$

where $\mathbf{I}_1$ and $\mathbf{I}_2$ are the transformed images. For the present example, the search range defined by Δd_{min} and Δd_{max} is $[-120, 50]$ and the block size b of the *sum of squared differences* (SSD) is 6.

The sub-pixel accuracy is achieved, similar to [17], by a refinement step in the least square sense on the basis of a Tayler expansion as follows

$$\Delta d_r = \Delta d_i + \frac{\sum_{i=-b}^{b} \mathbf{I}_2^{'}(u_t + i + \Delta d_i, v_t)\, \mathbf{e}(\Delta d_i, i)}{\sum_{i=-b}^{b} \mathbf{I}_2^{'}(u_t + i + \Delta d_i, v_t)^2} \qquad (20)$$

with

$$\mathbf{I}_2^{'}(u_t, v_t) = \mathbf{I}_2(u_t + 1, v_t) - \mathbf{I}_2(u_t, v_t). \qquad (21)$$

Finally, the coordinates of the corresponding feature point are defined by $\mathbf{m}_t^{'} = [u_t + \Delta d_r, v_t]^T$, where the x-coordinate is shifted by Δd_r and the y-coordinate remains the same.

E. Detection of mismatches

There are several interferences which lead to a wrong disparity estimation and consequently to mismatches. It is reasonable to detect such disturbances beforehand in order to prevent systematic errors in the estimation of the extrinsic parameters. The problems can be summarized as follows: shadow of the own vehicle or from other objects, no camera movement at all, reflections on the asphalt, detection of overtaking cars and objects besides the lane.

The detection of a static camera or shadow on the road is simple, since the disparity is zero or at least relatively small. The displacement of a feature point have to exceed a certain threshold

$$|\Delta d_{r,i}| > t_{hs}, \qquad (22)$$

otherwise it is rejected. If the vehicle moves forward, the disparity is naturally expected to be negative. A positive value indicates that an overtaking vehicle is in the focus of the camera. This type of error can be prevented as follows

$$\Delta d_{r,i} < 0. \qquad (23)$$

Reflections on the road surface or other types of misdetections have one in common: the resulting disparity differs substantially from all others. Consequently, only two feature pairs which have a similar displacement are taken into account. This is tested by the following criterion

$$|\Delta d_{r,i} - \Delta d_{r,j}| < t_{hr}. \qquad (24)$$

Finally, two feature pairs are selected from the set of feature pairs which fulfill the previous criteria. The feature points were extracted in the transformed version of the image and have to be converted into normalized coordinates as follows

$$\widetilde{\mathbf{m}}_{c,i} = \mathbf{R}_T \mathbf{V}^{-1} \widetilde{\mathbf{m}}_{t,i}. \qquad (25)$$

The normalized feature pairs are the starting point to estimate the extrinsic parameters as it will be proposed in Sec. III-G.

F. Parameterization of the rotation matrix

The question now is: How can the translation vector $\mathbf{t}_c$ be used to simplify the estimation of the homography matrix $\mathbf{H}_c$? We will realize this by a one-dimensional parameterization of the rotation matrix $\mathbf{R}_e$. Remember that a rotation matrix is normally parameterized by 3 DOF, i.e. one parameter for each rotation angle. Our basic idea is to exploit the fact that the motion of the camera in the CCS is defined by the translation vector $\mathbf{t}_c$ and the equivalent translation in the VCS is represented by the vector $\mathbf{t}_v$. This relationship is expressed by the equation

$$\mathbf{t}_c = \mathbf{R}_e \, \mathbf{t}_v, \tag{26}$$

see (47). The rotation matrix consists of three column vectors $\mathbf{R}_e = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$. Since $\mathbf{t}_v = [-\Delta s, 0, 0]^T$, the first column vector of $\mathbf{R}_e$ is proportional to $\mathbf{t}_c$ and can be calculated by $\mathbf{r}_1 = -\mathbf{t}_c / \|\mathbf{t}_c\|$.

The rotation matrix is orthonormal by definition and the last column vector $\mathbf{r}_3$ is perpendicular to the the first and the second one. Consequently, the vector $\mathbf{r}_3$ can be parameterized by the equation

$$\mathbf{r}_3(\alpha) = \cos(\alpha)\, \mathbf{n}_1 + \sin(\alpha)\, \mathbf{n}_2 \;\text{ with }\; \alpha \in [-\pi, \pi] \tag{27}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are two arbitrary unit vectors which are perpendicular to $\mathbf{r}_1$ and to each other. We choose

$$\mathbf{n}_1 \;=\; \frac{1}{\sqrt{r_{11}^2 + r_{21}^2}} \begin{bmatrix} -r_{21} \\ r_{11} \\ 0 \end{bmatrix} \text{ with } \mathbf{r}_1 = \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} \tag{28}$$

$$\mathbf{n}_2 \;=\; \mathbf{r}_1 \times \mathbf{n}_1. \tag{29}$$

This choice has the limitation that $\mathbf{r}_1 \neq [0, 0, 1]^T$. Therefore, a camera whose optical axis points exactly towards the direction of travel of the vehicle, is not permitted. This is never met for a side view camera application. For a front view camera application, the vectors should be chosen in a different manner to prevent numerical instabilities.

The value α is the remaining DOF in the one-dimensional parameter space of the rotation matrix, since the second column vector of $\mathbf{R}_e$ can be computed from the first and third column vector as follows

$$\mathbf{R}_e(\alpha) = [\mathbf{r}_1, -\mathbf{r}_1 \times \mathbf{r}_3(\alpha), \mathbf{r}_3(\alpha)]. \tag{30}$$

G. Estimation of the extrinsic parameters

Based on the assumptions in Sec. III-A, the rotation matrix $\mathbf{R}_v$ of homography matrix in (13) is replaced by the identity matrix. Furthermore, the parameterization of the rotation matrix $\mathbf{R}_e(\alpha)$ is applied. This leads to

$$\mathbf{H}_c(\theta, \alpha) = \mathbf{I} + \mathbf{R}_e(\alpha)\, \mathbf{\Theta}\, \mathbf{R}_e^T(\alpha) \tag{31}$$

with

$$\mathbf{\Theta} = \begin{bmatrix} 0 & 0 & \theta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \theta = \frac{\Delta s}{h_c}. \tag{32}$$

Note that the homography matrix is parameterized by θ and α. The previous equation can be reformulated as

$$\mathbf{H}_c(\theta, \alpha) = \mathbf{I} + \theta\, \mathbf{r}_1 \mathbf{r}_3^T(\alpha). \tag{33}$$

Besides the distance Δs, the extrinsic parameters $\mathbf{R}_e$ and h_c are the remaining components of the homography matrix. It is obvious that the camera height h_c cannot be estimated if the distance Δs is not known and vice versa. In this paper, we consider the case that Δs is unknown, but the camera height h_c is known. The camera height is obtained by an offline calibration method or directly from the measured installation height. Consequently, only the orientation of the camera has to be estimated. Fortunately, the estimation of the rotation matrix $\mathbf{R}_e$ stays unaffected from a wrong choice of h_c. This is due to the fact that θ is estimated anyway and the distance Δs is estimated implicitly from θ. Consequently, a wrong camera height will lead to a wrong distance Δs, but this is only of importance if the distance Δs is used at a later processing step.

In the following, we derive an analytic solution to determine the extrinsic parameters based on two feature pairs. Therefore, the parameterization $\mathbf{r}_3(\alpha)$ from (27) is substituted into (33) to get

$$\mathbf{H}_c(\alpha) = \mathbf{I} + \theta\, \mathbf{r}_1 (\cos(\alpha)\, \mathbf{n}_1^T + \sin(\alpha)\, \mathbf{n}_2^T). \tag{34}$$

According to (14), a single feature pair is related to the homography matrix as follows

$$\frac{1}{\lambda}\, \widetilde{\mathbf{m}}_c^{\prime} = \widetilde{\mathbf{m}}_c + \theta\, \mathbf{r}_1 \underbrace{(\cos(\alpha)\, \mathbf{n}_1^T + \sin(\alpha)\, \mathbf{n}_2^T)\, \widetilde{\mathbf{m}}_c}_{\mu} \tag{35}$$

where λ is the unknown arbitrary scale factor from (4). This factor can be solved easily with the last row of the previous SLE ($\frac{1}{\lambda} = 1 + \theta\, \mu\, r_{31}$). Substituted into the first and second row leads to two different equations for θ:

$$\theta_u = \frac{1}{\mu} \underbrace{\frac{u_c - u_c^{\prime}}{u_c^{\prime}\, r_{31} - r_{11}}}_{\eta_u} \;\text{ and }\; \theta_v = \frac{1}{\mu} \underbrace{\frac{v_c - v_c^{\prime}}{v_c^{\prime}\, r_{31} - r_{21}}}_{\eta_v}. \tag{36}$$

Actually, θ_u and θ_v, or η_u and η_v should be identical for a single feature pair because θ is defined by the distance Δs and the camera height h_c. The constraint that these values should be identical is equivalent to the epipolar constraint in (12). We consider the epipolar constraint during the extraction of the feature pairs. According to that, the value of η can be determined by one of the previous formulas, since $\eta = \eta_u = \eta_v$. Hence, the amount of parameters, which can be solved by the system of equations, is decreased. That is the reason why a single feature pair is insufficient to determine the parameter α if the value of θ is not known a priori. Contrary to [8] where θ is known, here two feature pairs are needed to solve α. Note that they are not allowed to be on the same epipolar line.

By definition, the value of θ should be independent of the choice of the feature pair, whereas μ and η do depend. Therefore, μ_1 and η_1 are the values from the first and μ_2 and η_2 from the second feature pair, respectively. These considerations lead to the equation

$$\mu_1\, \eta_2 - \mu_2\, \eta_1 = 0 \tag{37}$$

which can be reformulated as

$$A\, \cos(\alpha) + B\, \sin(\alpha) = \sqrt{A^2 + B^2}\, \sin(\alpha + \phi) = 0 \tag{38}$$

with

$$A = \;\; \eta_2\, \mathbf{n}_1^T\, \widetilde{\mathbf{m}}_{c,1} - \eta_1\, \mathbf{n}_1^T\, \widetilde{\mathbf{m}}_{c,2} \quad \text{and} \tag{39}$$

$$B = \;\; \eta_2\, \mathbf{n}_2^T\, \widetilde{\mathbf{m}}_{c,1} - \eta_1\, \mathbf{n}_2^T\, \widetilde{\mathbf{m}}_{c,2}. \tag{40}$$

There are two solutions for α: $-\phi$ and $\pi - \phi$, where $\tan(\phi) = A/B$. The sign is the only difference with respect to the resulting rotation vector: $\mathbf{r}_3(-\phi) = -\mathbf{r}_3(\pi - \phi)$. Only the correct solution is physically meaningful and can be easily identified. In general, it is a good idea to compute both $\mathbf{R}_e(-\phi)$ and $\mathbf{R}_e(\pi - \phi)$ to select the right one. A camera, for example, which is attached to the left side of the vehicle, fulfills $r_{32} > 0$, since the optical axis of the camera should point outwards. Alternatively, the constraint $\theta > 0$ can also indicate the correct solution since $\Delta s, h_c > 0$.

The final solution of $\mathbf{R}_e$ is converted into the Rodrigues notation [9] because the parameterization $\mathbf{R}_e(\alpha)$ is inconstant over time - due to the fact that the parameterization depends on $\mathbf{t}_c$. This is one possibility of an invertible representation of a rotation matrix. The three Rodrigues parameters are defined as follows

$$\omega_{rod} = \frac{\upsilon}{2\sin(\upsilon)}[r_{32} - r_{23}, r_{13} - r_{31}, r_{21} - r_{12}]^T \tag{41}$$

with

$$\upsilon = \arccos\left(\frac{\text{trace}(\mathbf{R}_e) - 1}{2}\right). \tag{42}$$

IV. Results

In the following, a complete video sequence of $60,000$ frames is processed to analyze the stability of the proposed calibration method. The camera operates with a resolution of 640x240 pixels and a frame rate of 30 fps.

Of course, we need the real extrinsic parameters as a reference for the evaluation of our system. Therefore, the reference parameters are obtained offline with a classical calibration method [1]. It uses a checkerboard pattern as calibration object and estimates the extrinsic parameter set.

A. Estimation of the essential matrix

The *Harris corner detector* in combination with a subsequent 16x16 SAD block matching extracts the required feature pairs for the estimation of the epipolar geometry. RANSAC subsequently estimates the essential matrix in the presence of mismatches. The robust estimation of the essential matrix consists of three parameters (cf. Sec. III-C), namely the components of the translation vector $\mathbf{t}_c$. The results of the estimation for the video sequence are illustrated in Fig. 7. Each color in the histogram plot represents the distribution of one component of the translation vector.

Each component has a noticeable peak at a certain value, which is very likely close to the real value of the translation vector. The final parameters are presented in Table I. They are obtained by a recursive filter, which approximates an average filter for each component of the translation vector with the aim to locate the peak in the distribution. The filter is adjusted in such a way that on the one hand it adopts slowly if the extrinsic parameter set has really changed and on the other hand it is robust against wrong estimates. In other words: the compromise between the adaptability and stability of the filter. Additionally, only those measurements are taken into account which are inside a certain window (see Fig. 7). The window is centered around the previous estimated value and the width is adjusted automatically, so that approximately 50% of the estimates are taken into account. In this way, wrong estimates, due to mismatches or a rotation of the vehicle, are rejected and do not corrupt the final result.

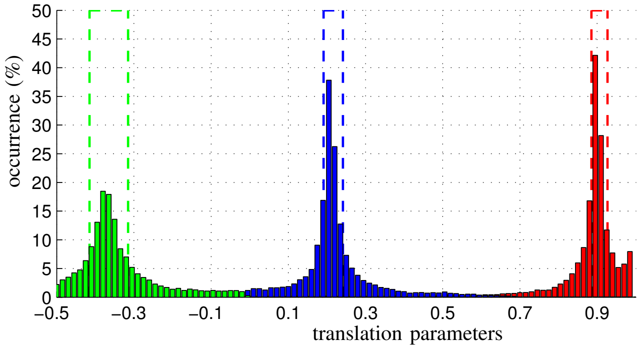

Fig. 7. The histogram over the three components of the translation vector $\mathbf{t}_c = [t_x \text{ (red, right)}, t_y \text{ (blue, middle)}, t_z \text{ (green, left)}]^T$, the adapting window for each parameter (dashed)

parameters of $\mathbf{t}_c$	(red)	(blue)	(green)
reference	0.9093	0.2081	-0.3603
measured	0.9059	0.2158	-0.3645

TABLE I

Final translation vector versus the reference

B. Estimation of the extrinsic rotation matrix

For the estimation of the extrinsic rotation matrix, we assume that the epipolar geometry was correctly estimated (see Table I) and initialize the required translation vector to $\mathbf{t}_c = [0.9059, 0.2158, -0.3645]^T$. The threshold $t_{hs} = 20$ was chosen, which represents the expected pixel displacement for a camera velocity of $30\,km/h$. This is no drawback since a vehicle normally travels much faster. Setting $t_{hr} = 20$ is also a good choice to reject the majority of mismatches, which would lead to systematic errors, but still accept most of the correctly assigned feature points. On average, every sixth image pair has two strong feature pairs which are expected to be correctly matched and suitable to estimate the rotation matrix. The results for the complete video sequence are illustrated in Fig. 8 as a histogram plot, while the rotation is represented in the Rodrigues notation.

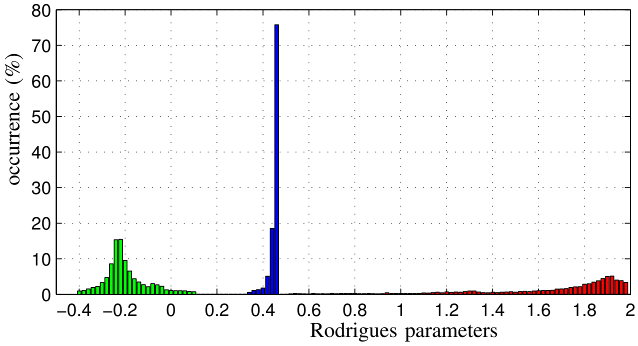

Fig. 8. The histogram over the three Rodrigues parameters $\omega_{rod} = [\omega_1 \text{ (red, right)}, \omega_2 \text{ (blue, middle)}, \omega_3 \text{ (green, left)}]^T$ of the rotation matrix with a total number of $10,000$ estimates

Each color in the histogram represents the distribution of one component in the Rodrigues notation. It is obvious that the distribution is different from each other. This can be explained by the parameterization of the rotation matrix, since only a subset of all possible rotations are parameterized. The final parameter set is computed in the same way as the peaks from the translation vector were selected. They are listed in Table II. The error between the final extrinsic rotation matrix $\mathbf{R}_{\text{fin}}$ and the reference matrix $\mathbf{R}_{\text{ref}}$ is $\arccos(\frac{1}{3}\text{trace}(\mathbf{R}_{\text{ref}}^T\mathbf{R}_{\text{fin}})) \approx 0.632°$.

parameters of ω_{rod}	(red)	(blue)	(green)
reference	1.9058	0.4542	-0.2172
measured	1.9057	0.4584	-0.2094

TABLE II

FINAL EXTRINSIC PARAMETERS VERSUS THE REFERENCE

V. CONCLUSION

We presented a new method to automatically calibrate the extrinsic parameters of a monocular camera. It basically exploits a homography between two camera shots if the road surface is visible and the camera has moved. A homography matrix has in general eight degrees of freedom. The definition of a reasonable vehicle coordinate system in combination with the epipolar geometry simplifies the parameterization of the underlying homography matrix. Furthermore, a similar process to the image rectification in stereo vision is introduced. This is the starting point for the extraction, selection and matching of potential feature points on the road surface. We derived an analytic solution to determine the extrinsic parameters based on two pairs of corresponding feature points.

Finally, we presented the results for a real-world video sequence. The final extrinsic parameters are estimated over various image pairs by a recursive filter which is robust against outliers. The resulting parameter set is competitive with the result obtained by the classical offline calibration method.

APPENDIX I

It is known in literature [9], [15] that a homography matrix for a plane Π_c is defined by

$$\mathbf{H}_c = \mathbf{R}_c + \mathbf{t}_c\,\mathbf{n}_c^T/d_c \tag{43}$$

where $[\mathbf{R}_c \;\; \mathbf{t}_c]$ is the Euclidean transform between two camera views in the CCS and the plane $\Pi_c : \mathbf{n}_c^T\mathbf{M}_c - d_c = 0$ is defined with respect to the first view. If $\mathbf{R}_e$ and $\mathbf{t}_e$ do not change between these two camera views, $[\mathbf{R}_c \;\; \mathbf{t}_c]$ can be determined by combining (2), (5), and (7):

$$\mathbf{M}_c^{'} = \underbrace{\mathbf{R}_e\mathbf{R}_v\mathbf{R}_e^T}_{\mathbf{R}_c}\mathbf{M}_c + (\underbrace{-\mathbf{R}_e\mathbf{R}_v\mathbf{R}_e^T\mathbf{t}_e + \mathbf{R}_e\mathbf{t}_v + \mathbf{t}_e}_{\mathbf{t}_c}). \tag{44}$$

A plane $\Pi_v : \mathbf{n}_v^T\mathbf{M}_v - d_v = 0$ defined in the VCS is transformed to a plane in the CCS by

$$\Pi_c : \underbrace{\mathbf{n}_v^T\mathbf{R}_e^T}_{\mathbf{n}_c^T}\mathbf{M}_c - (\underbrace{d_v + \mathbf{n}_v^T\mathbf{R}_e^T\mathbf{t}_e}_{d_c}) = 0. \tag{45}$$

With the definition of the VCS from Sec. II-B, the road surface is defined by $\mathbf{n}_v = [0, 0, 1]^T$ and $d_v = 0$. The assumption that the rotation matrix $\mathbf{R}_v$ has the form

$$\mathbf{R}_v = \begin{bmatrix} \cos(\Delta\omega) & \sin(\Delta\omega) & 0 \\ -\sin(\Delta\omega) & \cos(\Delta\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{46}$$

and the definition of the extrinsic parameters in (6) lead to

$$\begin{aligned} \mathbf{t}_c &= \mathbf{R}_e(\mathbf{R}_v\mathbf{t}_h - \mathbf{t}_h + \mathbf{t}_v) &&= \mathbf{R}_e\mathbf{t}_v, && (47) \\ d_c &= -\mathbf{n}_v^T\mathbf{R}_e^T\mathbf{R}_e\mathbf{t}_h &&= -h_c. && (48) \end{aligned}$$

Substituted into (43), we finally obtain the homography matrix in (13).

REFERENCES

[1] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, 1998.

[2] R. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[3] S. Hold, C. Nunn, A. Kummert, and S. Müller-Schneiders, “Efficient and robust extrinsic camera calibration procedure for lane departure warning,” in *IEEE Proc. Intelligent Vehicles Symposium*, 2009, pp. 382–387.

[4] M. Bellino, Y. Kolski, and S. Jacot, “Calibration of an embedded camera for driver-assistant systems,” in *IEEE Proc. International Conference on Intelligent Transportation Systems*, 2005, pp. 354–359.

[5] H.-J. Lee and C.-T. Deng, “Camera models determination using multiple frames,” in *IEEE Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 127–132.

[6] M. Wu and X. An, “An automatic extrinsic parameter calibration method for camera-on-vehicle on structured road,” in *IEEE Proc. International Conference on Vehicular Electronics and Safety*, 2007, pp. 1–5.

[7] S. Hold, S. Görmer, A. Kummert, M. Meuter, and S. Müller-Schneiders, “A novel approach for the online initial calibration of extrinsic parameters for a car-mounted camera,” in *IEEE Proc. International Conference on Intelligent Transportation Systems*, 2009, pp. 420–425.

[8] M. Miksch, B. Yang, and K. Zimmermann, “Homography-based extrinsic self-calibration for cameras in automotive applications,” in *Workshop on Intelligent Transportation*, 2010, pp. 17–22.

[9] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[10] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[12] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] Z. Zhang, “Determining the epipolar geometry and its uncertainty: A review,” *International Journal of Computer Vision*, vol. 27, no. 2, pp. 161–195, 1998.

[14] X. Armangué and J. Salvi, “Overall view regarding fundamental matrix estimation,” *Image and Vision Computing*, vol. 21, pp. 205–220, 2003.

[15] O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[16] R. I. Hartley, “In defense of the eight-point algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[17] S. Birchfield, “Derivation of Kanade-Lucas-Tomasi tracking equation,” 1997.