Leveraging scale- and orientation-covariant features for planar motion estimation

Marcus Valtonen Örnhag and Alberto Jaenal

Ericsson Research, Sweden
{marcus.valtonen.ornhag,alberto.jaenal.galvez}@ericsson.com

Abstract. In this paper, we derive a linear constraint for planar motion leveraging scale- and orientation-covariant features, *e.g.*, SIFT, which is used to create a novel minimal solver for planar motion requiring only a single covariant feature. We compare the proposed method to traditional point-based solvers and solvers relying on affine correspondences in controlled synthetic environments and well-established datasets for autonomous driving. The proposed solver is integrated into a modern robust estimation framework, where it is shown to accelerate the complete estimation pipeline more than 25×, compared to state-of-the-art affine-based minimal solvers, with negligible loss in precision[1].

Keywords: Planar motion · covariant features · epipolar geometry · polynomial solvers

1 Introduction

Relative pose estimation is a vital component in many core computer vision applications, including Simultaneous Localization and Mapping (SLAM) [13], Visual Odometry (VO) [42] and Structure-from-Motion (SfM) [43]. Such frameworks typically start by finding and extracting image descriptors, which are in turn matched between images. Subsequently, the matched image descriptors are used to estimate the geometry using a robust matching framework such as Random Sample Consensus (RANSAC) [24]. In RANSAC, a small subset of the matches is used to estimate the geometry, and the hypothesis is tested against the complete (or a larger) set of correspondences to determine potential outliers. This process is repeated with a new hypothesis, and the hypothesis with the most inliers is finally chosen. The number of necessary iterations to select an all-inlier set is exponentially dependent on the sample size, and, as such, it is desirable to work with *minimal solvers*. Famous examples of minimal solvers for relative pose estimation are, *e.g.*, the 7-point algorithm [44] for estimating the fundamental matrix and the 5-point algorithm [36] for estimating the essential matrix.

In many practical scenarios, physical limitations may restrict the degrees of freedom of the camera movement, thus constraining it to a certain type of

[1] Code available here: https://github.com/EricssonResearch/eccv-2024

motion. This is often the case in autonomous navigation, where one or more cameras are rigidly mounted on a vehicle, which can therefore only rotate in one direction, while traveling in a plane, relative to the vehicle [14, 31, 41]. While in reality, the motion model might not hold exactly, it is still a good approximation thereof, and modern RANSAC-like frameworks, *e.g.*, LO-RANSAC [15] and more recently GC-RANSAC [6], apply non-linear refinement within the framework to deal with possible small deviations. This makes the application of such models feasible in practice; however, some authors make additional assumptions aimed at improving performance in the presence of such imperfections [17,49,50]. In general, enforcing motion models tailored to the application causes a reduction in the number of degrees of freedom of the motion, and consequently, in the number of point correspondences required by the solver. This might entail, *e.g.*, adding planarity assumptions on the scene, thus enabling the estimation of a homography with fewer points [22, 39, 40]; or, applying homographies to pure rotation [12]; or, adding auxiliary sensory information, such as IMU data, to align the poses, as in [19–21, 25, 46–48, 52].

Depending on the type of image descriptor used in the pipeline, different kinds of geometrical information might be available. A common practice has been to exclusively use point correspondences to estimate the geometry, thus neglecting additional sources of information embedded in the image descriptors. In recent years, affine correspondences (ACs) have been proposed as an alternative, and with them a range of algorithms [4, 5, 7, 18, 26, 29]. An affine correspondence $(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{A})$ is a triplet where $\boldsymbol{p}_1 \leftrightarrow \boldsymbol{p}_2$ are the corresponding homogeneous points and $\boldsymbol{A}$ is a local affine transformation represented by a 2×2 matrix

$$\boldsymbol{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} . \tag{1}$$

Such features can be extracted using, *e.g.*, ASIFT [35] or AffNet [34]. When using point-based features, only a single linear constraint in the elements of the fundamental matrix is obtained from the epipolar constraint [32]; however, it has been shown that a single AC gives rise to three linear constraints [9, 37]. Thus, only three ACs are required to estimate the fundamental matrix [9] and two ACs for the essential matrix [3, 23] or an homography [37]. In [30] it was shown that only a single AC is necessary to estimate the relative pose under the planar motion assumption or with knowledge of a vertical direction, while a very similar affine solver for planar motion was proposed in [31]. The generalized relative pose problem was treated in [28] and a globally optimal solver using affine correspondences to estimate the relative pose of two views with a known vertical direction in [51]. In [8] many issues regarding practical applications of using affine features were addressed, and a set of guidelines were presented on how to achieve similar accuracy with affine solvers compared to traditional point-based solvers while benefiting from faster run times.

With the advancements and benefits of using affine features for camera geometry estimations, a surge in leveraging the inherent information provided by image descriptors has gained attention in the community. Among them is the

idea of approximating affine features using the information provided in scale- and orientation-covariant features, presented in [2]. Early work includes homography estimation from two SIFT features [4], showing that each SIFT feature can yield two additional linear equations. In [5], a new constraint, leveraging scale- and orientation-covariant features, was introduced for estimating the fundamental matrix, and variations thereof. It adds a second linear equation in the elements of the fundamental matrix for every point pair, hence reducing the number of necessary points from N to $\lceil \frac{N}{2} \rceil$ SIFT features. In practice, this implies that an epipolar geometry problem requiring N point correspondences can potentially be solved using $\lceil N/2 \rceil$ SIFT correspondences, and it was shown that the fundamental matrix can be estimated from four SIFT correspondences, and the essential matrix from three. In [1] the relationship between surface normals, camera projections, and affine transformation was studied. From this study it was proposed that the local affine transformation could be defined as the product of the Jacobians of the projection functions with respect to the image directions [4]

$$\boldsymbol{J} = \boldsymbol{J}_2 \boldsymbol{J}_\mathbf{1}^{-1}, \tag{2}$$

where $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$ are the Jacobians of the 3D $\rightarrow$ 2D projection functions for the first and second image, respectively. Furthermore, the Jacobian matrices can be decomposed as $\boldsymbol{J}_i = \boldsymbol{R}_i \boldsymbol{U}_i$, where $\boldsymbol{R}_i$ is a rotation matrix corresponding to the orientation of the SIFT feature, and $\boldsymbol{U}_i$ contains information about the scales and shear. It was shown in [5], that this may be expressed conveniently as

$$\boldsymbol{A} = \boldsymbol{R}_2 \boldsymbol{U} \boldsymbol{R}_1^T = \begin{bmatrix} \cos(\alpha_2) & -\sin(\alpha_2) \\ \sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \begin{bmatrix} q_u & w \\ 0 & q_v \end{bmatrix} \begin{bmatrix} \cos(\alpha_1) & -\sin(\alpha_1) \\ \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix}^T, \tag{3}$$

where q_u and q_v is the combined scale along the axes u and v, and w is the corresponding shear. This formulation allows for a compact and useful description of the relation between affine and SIFT correspondences.

The main drawback of affine features is the slower extraction time, which can greatly reduce its applicability. In [35] the complexity for computing ASIFT features is reported to be 13.5 times the complexity for computing SIFT features. Furthermore, the complexity of ASIFT descriptor matching is $13.5^2 \approx 180$ times greater than that of SIFT. While later work has improved these times, *e.g.*, using GPU acceleration [16], or through learning-based affine descriptors [33, 34], the total extraction time is still significantly higher than for regular image descriptors.

In this paper, we will explore the benefits of approximating the affine features directly from the scale and orientation intrinsically encoded in the covariant feature descriptors applied to planar motion. The main contributions of this work are threefold:

- A fast novel minimal solver using only a single scale- and orientation-covariant correspondence.
- We show that the preprocessing overhead of extracting and matching affine features is a real concern when integrated into a state-of-the-art robust matching framework.

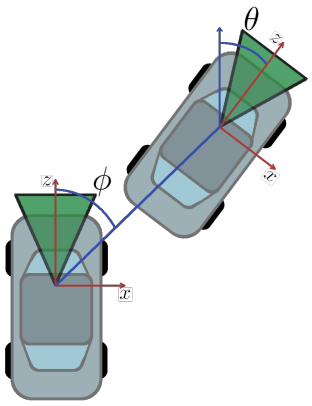

Fig. 1: A camera is rigidly mounted on top of a vehicle. The motion is constrained to a rotation about the y-axis by an angle θ and a translation $\boldsymbol{t} = [t_x, 0, t_z]^T$. The magnitude of the translation vector cannot be obtained, only the direction, which is represented by the angle ϕ. Consequently, the relative planar motion problem has two degrees of freedom.

– A thorough discussion on the differences between point-based, SIFT-based, and affine correspondence-based planar motion estimation supported by theoretical analysis and experimental validation.

2 Epipolar geometry for planar motion

In this section, we present the problem, derive the planar motion SIFT constraint, and use it to build a novel minimal solver. Furthermore, we discuss how to normalize the image coordinates to improve numerical stability.

2.1 Problem formulation

Consider a forward-facing camera mounted rigidly on a vehicle traveling along a planar surface. We may choose the coordinate system such that the camera is moving parallel to the plane $y = 0$, hence any translation of the camera can be expressed as $\boldsymbol{t} = [t_x, 0, t_z]^T$. Furthermore, since the camera is forward-facing, the only rotation that may occur is about the y-axis, modelled by a rotation matrix $\boldsymbol{R}_y$. We seek to estimate the relative motion; hence, the translation vector $\boldsymbol{t}$ can only be recovered up to scale. Therefore, only the direction of the translation can be obtained and the problem can be parameterized as

$$\boldsymbol{R}_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \qquad \boldsymbol{t} = -\boldsymbol{R}_y \begin{bmatrix} \rho\sin\phi \\ 0 \\ \rho\cos\phi \end{bmatrix}, \tag{4}$$

where the translation scale ρ is unknown, θ is the yaw angle and ϕ the direction of the translation, see Fig. 1. Consequently, the relative planar motion problem has two degrees of freedom. The corresponding essential matrix is given by $\boldsymbol{E} = \boldsymbol{R}_y [\boldsymbol{t}]_\times$, or explicitly,

$$\boldsymbol{E} = \rho \begin{bmatrix} 0 & \cos(\theta - \phi) & 0 \\ -\cos\phi & 0 & \sin\phi \\ 0 & \sin(\theta - \phi) & 0 \end{bmatrix} . \tag{5}$$

Equivalently, we may express the essential matrix as

$$\boldsymbol{E} = \begin{bmatrix} 0 & e_1 & 0 \\ e_2 & 0 & e_3 \\ 0 & e_4 & 0 \end{bmatrix} . \tag{6}$$

The well-known rank constraint $\det(\boldsymbol{E}) = 0$ is trivially fulfilled, and the trace constraint $2\boldsymbol{E}\boldsymbol{E}^T\boldsymbol{E} - \mathrm{tr}(\boldsymbol{E}\boldsymbol{E}^T)\boldsymbol{E} = 0$ is reduced to a single equation

$$e_1^2 - e_2^2 - e_3^2 + e_4^2 = 0 . \tag{7}$$

Using only point-based information, a single linear equation in the elements e_i can be obtained using the epipolar constraint between two views [32]. Given two corresponding homogeneous points $\boldsymbol{p_1} \leftrightarrow \boldsymbol{p}_2$, it is given by $\boldsymbol{p}_2^T \boldsymbol{E} \boldsymbol{p}_1 = 0$. For the relative planar motion case, this is reduced to

$$v_1 u_2 e_1 + u_1 v_2 e_2 + v_2 e_3 + v_1 e_4 = 0 . \tag{8}$$

where $\boldsymbol{p}_1 = [u_1,\, v_1,\, 1]^T$ and $\boldsymbol{p}_2 = [u_2,\, v_2,\, 1]^T$.

2.2 Affine transformations and the planar motion model

As was previously mentioned, this paper aims to incorporate more constraints from the information embedded in the image descriptors. In order to do so, we will go through the approximation of affine transformations, and as such it is necessary to understand the relationship between the fundamental matrix and an affine transformation. Consider therefore the essential matrix $\boldsymbol{E}$ compatible with the planar motion constraints as parameterized in (6) and an affine correspondence $(\boldsymbol{p}_1,\, \boldsymbol{p}_2,\, \boldsymbol{A})$. Let $\boldsymbol{p}_i = [u_i,\, v_i,\, 1]^T$, then the corresponding epipolar lines are given by

$$\boldsymbol{l}_1 = \boldsymbol{E}^T \boldsymbol{p}_2 = \begin{bmatrix} v_2 e_2 \\ u_2 e_1 + e_4 \\ v_2 e_3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{l}_2 = \boldsymbol{E}\boldsymbol{p}_1 = \begin{bmatrix} v_1 e_1 \\ u_1 e_2 + e_3 \\ v_1 e_4 \end{bmatrix} . \tag{9}$$

In [9], it was found that that $\boldsymbol{A}$ transforms $\boldsymbol{v}_1$ to $\boldsymbol{v}_2$, where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the directions of the epipolar lines of unit length in the first and second images, respectively. Furthermore, this implies that $\boldsymbol{A}\boldsymbol{v}_1 \parallel \boldsymbol{v}_2$, which in turn can be expressed as [5]

$$\boldsymbol{A}^T \boldsymbol{n}_1 = -\boldsymbol{n}_2, \tag{10}$$

where $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are the normals of unit length of the corresponding epipolar lines, respectively. These are simply the first two entries of the vectors $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$, respectively. Combining (9) and (10) yields two equations that are linear in e_i

$$a_1v_1e_1 + a_3u_1e_2 + v_2e_2 + a_3e_3 = 0, \tag{11}$$
$$(a_2v_1 + u_2)e_1 + a_4u_1e_2 + a_4e_3 + e_4 = 0\ . \tag{12}$$

These constraints are a special case of the ones reported in [5], where the general fundamental matrix was treated. We will use them to derive a planar motion constraint for SIFT features in the following section.

2.3 Planar motion SIFT constraint

Scale- and orientation-covariant features contain information about the scale q_i and the orientation α_i. In [4, 5] it was found that there are three constraints connecting an affine correspondence $\boldsymbol{A}$ and the orientation and scale of a corresponding SIFT features. These are given by

$$a_2a_3 - a_1a_4 + q^2 = 0, \tag{13}$$
$$a_3c_1 + a_4s_1 - s_2q = 0, \tag{14}$$
$$a_1c_1 + a_2s_1 - c_2q = 0, \tag{15}$$

where $q = q_2/q_1$, $c_i = \cos\alpha_i$ and $s_i = \sin\alpha_i$. In order to generate a planar motion-compatible constraint leveraging SIFT features, we will use these and the constraints provided in the previous section. To this end, consider the ideal J created by the polynomials (13)–(15) and (11)–(12). To eliminate the affine correspondence, we compute the corresponding elimination ideal given by $J_1 = J \subset \mathbb{C}[e_1, e_2, e_3, e_4, u_1, v_1, u_2, v_2, q, c_1, s_1, c_2, s_2]$, which is generated by

$$(v_1qc_2 + u_2s_1)e_1 + (u_1qs_2 + v_2c_1)e_2 + qs_2e_3 + s_1e_4 = 0\ . \tag{16}$$

This constraint encodes the relationship between SIFT correspondences and the relative pose problem under the planar motion assumption. In the next section, we will use this constraint to build a minimal solver using a single SIFT correspondence.

2.4 A minimal 1 SIFT planar motion solver

Consider the parameterization of the epipolar matrix as defined in (6). Given a single SIFT correspondence, two equations are obtained that are linear in e_i, from (8) and (16), respectively. We may write these as a system of equations

$$\boldsymbol{M}\boldsymbol{e} = 0, \tag{17}$$

where $\boldsymbol{M}$ is a 2×4 matrix. This implies that the matrix $\boldsymbol{M}$ has a two-dimensional nullspace $\boldsymbol{N}$, which can be parameterized as

$$\boldsymbol{N} = \alpha\boldsymbol{n}_1 + \beta\boldsymbol{n}_2\ . \tag{18}$$

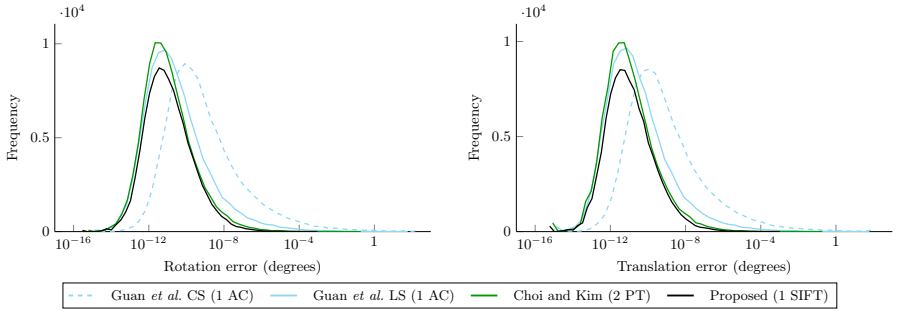

Fig. 2: *Numerical stability.* The frequencies (100 000 runs; vertical axis) of $\log_{10}$ angular errors (in degrees; horizontal axis) for the yaw angle and the translational component on noise-free data estimated by the two-point (2PT) solver [14], two 1 AC solvers proposed in [30] and the proposed SIFT-based solver.

We may fix the scale by letting $\alpha = 1$. Hence, given a matched pair of SIFT correspondences, we may form the matrix $\boldsymbol{M}$ and compute the null space vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$. By inserting these into the trace constraint (7) a quadratic equation in the single unknown β is obtained. The two roots β_1 and β_2 can be computed analytically and the final solution is obtained by backward substituting them in (18).

2.5 A note on normalization

It is well-known that numerical stability is increased by normalizing the point coordinates. This is still possible with SIFT features. The formula for normalizing $\boldsymbol{A}$ is as follows [3]:

$$\hat{\boldsymbol{A}} = \boldsymbol{K}_2 \begin{bmatrix} \boldsymbol{A} & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{K}_1^{-1} = \frac{f_2}{f_1} \boldsymbol{A}, \tag{19}$$

where $\boldsymbol{K}_1$ and $\boldsymbol{K_2}$ are the calibration matrices for the first and second cameras, respectively. The equations relating affine features and SIFT features (13)–(15) are then transformed to

$$\hat{a}_2 \hat{a}_3 - \hat{a}_1 \hat{a}_4 + \hat{q}^2 = 0, \tag{20}$$

$$\hat{a}_3 c_1 + \hat{a}_4 s_1 - s_2 \hat{q} = 0, \tag{21}$$

$$\hat{a}_1 c_1 + \hat{a}_2 s_1 - c_2 \hat{q} = 0, \tag{22}$$

where $\hat{q} = f_2 q / f_1$. It should be noted that the structure of the constraints is not changed, merely the coefficients in the template matrix. Furthermore, for $\boldsymbol{K}_1 = \boldsymbol{K}_2$ the constraints are unaffected.

3 Experiments

We perform a comprehensive review of planar motion-compatible solvers utilizing point-based, SIFT, and affine correspondences, listing their respective strengths

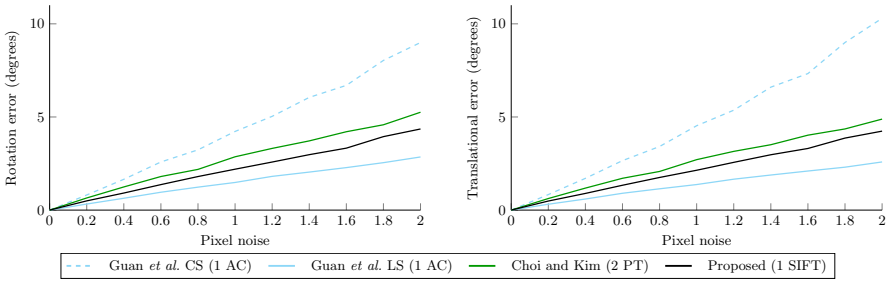

Fig. 3: *Sensitivity to noise (general motion).* Relative planar pose estimation in a synthetic environment. The median angular errors (vertical axis; in degrees) of the estimated rotation and translation are shown as functions of the standard deviation of the pixel noise (horizontal axis), averaged over 10 000 runs per noise level. The compared methods are the two affine (1 AC) solvers in [30], the 2-point solver in [14], and the proposed 1-SIFT solver.

and weaknesses. To realize this, we use the two-point solver by Choi and Kim [14] (2 PT), the solvers utilizing a single affine correspondence by Guan *et al.* [30] (1 AC), and the proposed 1 SIFT solver. First, we consider a controlled synthetic environment, before continuing with well-established real automotive datasets. In all evaluations, the point coordinates are normalized using the intrinsic calibration matrix. As described in Sec. 2.5, this does not affect the solver when $\boldsymbol{K}_1 = \boldsymbol{K}_2$ and it is known that this also holds for affine correspondences [3].

The error metrics used to estimate the performance of the different solvers compare the angular differences of the rotation estimate and the translation estimate, with respect to the ground truth, separately. Specifically

$$\epsilon_{\boldsymbol{R}} = \arccos\left(\frac{\operatorname{tr}(\boldsymbol{R}_{\text{gt}}\boldsymbol{R}^T) - 1}{2}\right), \tag{23}$$

$$\epsilon_{\boldsymbol{t}} = \arccos\left(\frac{\boldsymbol{t}^T\boldsymbol{t}_{\text{gt}}}{\|\boldsymbol{t}\| \cdot \|\boldsymbol{t}_{\text{gt}}\|}\right) . \tag{24}$$

Again, note that the length of the translation cannot be obtained, so the metric $\epsilon_{\boldsymbol{t}}$ accounts for the angular difference between estimated $\boldsymbol{t}$ and the ground truth $\boldsymbol{t}_{\text{gt}}$.

All solvers are implemented in C++ using the Eigen linear algebra package and integrated with `pygcransac`[2]. The two-point solver is already available in the framework, and the AC-based methods are ported directly from the MATLAB implementation provided by the authors[3].

3.1 Synthetic Evaluation

In order to compare the proposed solver with the existing state-of-the-art solvers we create a synthetic scene consisting of two calibrated cameras $\boldsymbol{P}_1 = \boldsymbol{K}[\boldsymbol{I}\ 0]$

[2] `https://github.com/danini/graph-cut-ransac`
[3] `https://github.com/jizhaox/relative_pose_from_affine`

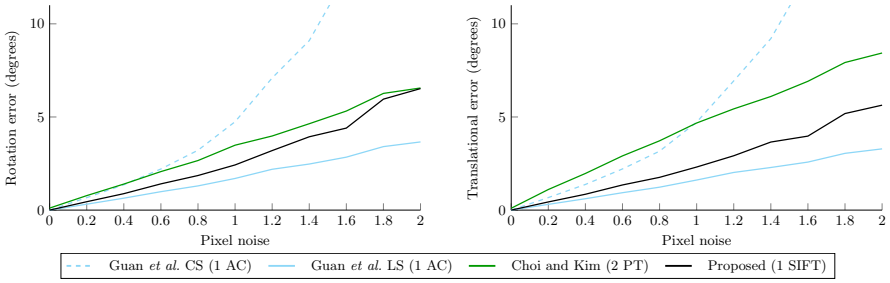

Fig. 4: *Sensitivity to noise (forward motion).* Relative planar pose estimation in a synthetic environment. The median angular errors (vertical axis; in degrees) of the estimated rotation and translation are shown as functions of the standard deviation of the pixel noise (horizontal axis), averaged over 10 000 runs per noise level. The compared methods are the two affine (1 AC) solvers in [30], the 2-point solver in [14], and the proposed 1-SIFT solver.

and $\boldsymbol{P}_1 = \boldsymbol{K}[\boldsymbol{R_y}\ \boldsymbol{t}]$. Random 3D points $\boldsymbol{X}_i$ are then generated in front of the cameras such that $\boldsymbol{X}_i = [x_i, y_i, z_i]^T$ is within a box $x_i \in [-5, 5]$, $y_i \in [-5, 5]$, and $z_i \in [10, 20]$. The length of the translation vector $\boldsymbol{t}$ is set to two units, and the angle of the translation ϕ and the rotation θ are randomly selected in the interval $[-30°, 30°]$. The intrinsic matrix $\boldsymbol{K}$ encodes a fixed focal length $f = 1000$ and a principal point $(u_0, v_0) = (320, 240)$, simulating an image size of 640×480 pixels. For each $\boldsymbol{X}_i$, four additional points are generated on a planar surface in the neighborhood of it and used to estimate a homography, which in turn is used to calculate the affine transformation associated with it, as discussed in [5]. This allows one to add noise to the affine correspondences in a controlled and interpretable manner. Finally, we generate SIFT correspondences by utilizing the decomposition from (3), as in [5]. Since the decomposition is not unique, random values were assigned to $\boldsymbol{J}_1$, thus allowing us to compute $\boldsymbol{J_2} = \boldsymbol{A}\boldsymbol{J}_1$.

We test several aspects, including: numerical stability, the sensitivity to pixel noise, forward motion, and nonplanar motion.

Numerical stability. In Fig. 2, the numerical stability of the minimal solvers is tested on noise-free data. All methods are performing well and are numerically accurate. Only the linear 1 AC method is deviating from the rest and this is likely due to the method not enforcing the trace constraint (7), since, the problem is overdetermined with a single AC constraint, which is why the least-squares (LS) solver was proposed in [30]. The LS solver, however, is significantly more computationally demanding, which is shown in Tab. 1.

Sensitivity to noise. By adding zero-mean Gaussian noise with varying standard deviation to the point coordinates as well as the coordinates defining the affine transformations—and therefore the SIFT features—we may compare the solvers' respective sensitivity in a controlled environment. Again, the linear 1 AC solver by Guan *et al.* [30], which does not enforce the trace constraint (7) is infe-

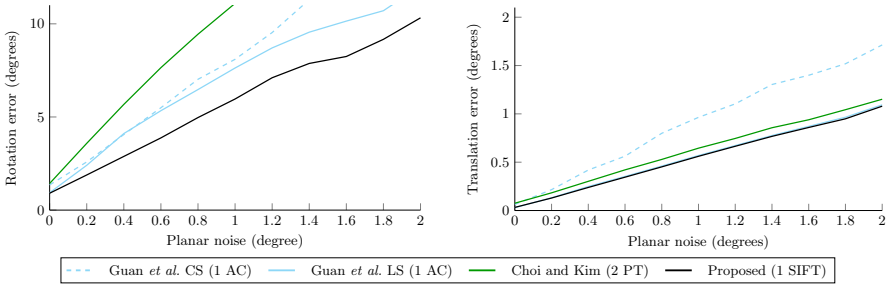

Fig. 5: *Nonplanar motion.* Relative planar pose estimation in a synthetic environment. The median angular error (vertical axis; in degrees) of the estimated rotation and translation is shown. The errors of the minimal solvers are shown as functions of the nonplanar noise (horizontal axis; in degrees), averaged over 10 000 runs per noise level.

rior among the compared methods. The proposed 1-SIFT method is performing slightly better than the 2-point method by Choi and Kim [14], both of which take the trace constraint into consideration. The least-squares 1 AC solver by Guan *et al*. [30] shows the best performance, and incorporates all affine constraints available, as well as the trace constraint.

Forward motion. In Fig. 4, we show a purely forward motion where $\phi = 0$ and there is a small random rotation $\theta \in [-5°, 5°]$, following the setup in [31]. This is a common situation in autonomous driving. The effect of enforcing the trace constraint is even more noticeable, as the linear 1 AC solver is diverging from the others after more than 1 pixel of noise.

Nonplanar motion. In real situations, the planar motion model is only approximate. This can be caused by the terrain, *e.g*., due to hills, and other nonplanar roads, but also incorrectly mounted cameras or vibrations, distorting the viewing angle which is assumed to be traveling in a plane. Hence, it is of importance that the solvers are stable with respect to these types of perturbations. Therefore, we corrupt the rotational component with Gaussian noise by adding a small rotational component about the x- and z-axis. Furthermore, we simulate a small hill by adding a positive contribution to the translational component t_y corresponding to the planar noise, similar to [31]. In Fig. 5 we show how the minimal solvers are affected. The proposed 1-SIFT solver and the 1 AC LS solver by Guan *et al*. [30] are performing similarly for the translational component, but the proposed SIFT-based solver performs better for the rotational component.

Timings. The proposed 1 SIFT solver is comparatively fast. We measure the computational time by averaging 100 000 executions of random input data on a desktop computer equipped with a 13th Gen Intel(R) Core(TM) i9-13900K @ 3.00GHz. The results are shown in Tab. 1. It should be noted that the 1 AC LS solver, which is the main competitor to the proposed 1 SIFT solver in the

synthetic experiments, is more than 50× as computationally heavy compared to the proposed 1 SIFT solver.

Table 1: Timings of the minimal solvers (in nanoseconds) averaged over 100 000 runs. Best marked in **bold**.

1 AC (CS) [30]	1 AC (LS) [30]	2 PT [14]	1 SIFT (Proposed)
793	12 427	246	**177**

3.2 Real experiments

In order to evaluate the performance of the different solvers on real data, we rely on two well-known automotive datasets: KITTI [27] and Malaga Urban [10], both collected using a car navigating in urban environments with a forward-facing stereo camera, among other sensors.

To extract the image features for the point and the SIFT cases, we employed the OpenCV implementation [11] of SIFT. For the case of affine features, AffNet [34] was used, executed on top of the Difference of Gaussians detector implemented in OpenCV and HardNet descriptor [33] from Kornia [38], which requires GPU acceleration for the feature extraction. Lowe's ratio test was carried out to establish putative matches with the default thresholds of 0.8 and 0.9, respectively, for the cases of SIFT and affine features. The compared methods are the five point estimator [36] (5PC), the Guan CS solver [30] (1AC), the 2-point solver in [14] (2PC), and the proposed 1-SIFT solver. As a robust estimator, we used the state-of-the-art RANSAC framework, Graph-Cut RANSAC (GC-RANSAC) [6], which is a locally optimized RANSAC alternating graph-cut and model re-fitting in the local optimization step. The minimal solvers considered in this paper were integrated into the framework, apart from Guan's LS method, due to its excessive runtime, which makes it infeasible for practical applications. The experiments were conducted on a desktop computer equipped with a 13th Gen Intel(R) Core(TM) i9-13900K @ 3.00GHz and an NVIDIA GeForce RTX 4090 GPU.

KITTI Dataset For the KITTI dataset [27] we exclusively use the left camera for the relative pose estimation problem. The ground truth poses are provided for sequences 00 to 10 from GPS and IMU data, and the overall dataset is well-approximated by the planar motion model. In total, this amounts to approximately 23 000 images, which are used for the evaluation using image pairs with stride 10. For the non-minimal solver, we use the Stewenius [45] five-point algorithm, which does not enforce the planar motion model. This, however, is beneficial for the KITTI data, as the motion is only approximately planar due to non-planar roads, etc. Following the setup used in [30], the inlier-outlier threshold is set to 2 pixels and is normalized by the focal lengths. Furthermore, the minimal number of RANSAC iterations is set to 100, and all other parameters use the default settings provided by the GC-RANSAC framework.

Table 2: Results on the KITTI dataset. Best values marked in **bold** and second best in *italic*. Errors are expressed in degrees.

Seq	Nister (5PC)				Choi (2PC)				Guan (1AC)				Our (1SIFT)			
	ϵ_t		ϵ_R		ϵ_t		ϵ_R		ϵ_t		ϵ_R		ϵ_t		ϵ_R	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
00	1.234	0.701	0.341	0.130	1.276	0.699	0.336	0.129	**1.089**	**0.674**	**0.291**	**0.120**	*1.225*	*0.687*	*0.286*	*0.126*
01	*4.583*	0.875	1.333	*0.145*	5.695	0.890	*1.117*	0.151	4.788	**0.850**	1.733	0.160	**4.002**	*0.869*	**0.774**	**0.140**
02	1.758	0.730	0.532	0.136	1.452	0.726	0.553	0.135	*0.958*	*0.674*	**0.329**	**0.116**	**0.955**	**0.669**	*0.374*	*0.120*
03	0.764	0.533	0.076	0.063	0.761	0.531	0.074	*0.062*	**0.745**	**0.507**	**0.071**	**0.059**	*0.756*	*0.511*	**0.071**	*0.062*
04	*0.988*	0.436	*0.239*	0.094	1.971	0.431	0.927	0.093	**0.427**	**0.392**	**0.111**	**0.084**	0.996	*0.423*	0.829	**0.084**
05	2.715	0.417	*0.147*	0.090	2.783	0.412	0.158	0.090	*2.583*	*0.394*	0.257	*0.085*	**2.581**	**0.384**	**0.144**	**0.083**
06	0.659	0.385	*0.212*	*0.092*	0.794	0.388	0.352	0.094	*0.627*	**0.360**	0.467	*0.092*	**0.604**	*0.384*	**0.180**	**0.091**
07	5.815	0.620	0.442	0.104	5.834	0.629	0.470	0.104	**5.494**	**0.590**	**0.133**	*0.102*	*5.547*	*0.608*	*0.350*	**0.099**
08	2.416	0.839	0.234	0.111	2.330	0.830	*0.207*	0.111	*2.153*	**0.799**	0.221	**0.103**	**2.136**	*0.820*	**0.196**	*0.109*
09	1.374	0.556	0.388	0.147	1.760	0.564	0.426	0.150	**0.845**	*0.475*	**0.205**	**0.122**	*1.058*	**0.466**	*0.295*	*0.142*
10	1.211	0.591	0.265	0.114	1.580	0.592	0.426	0.114	**0.943**	**0.516**	**0.198**	**0.109**	*1.136*	**0.516**	*0.252*	*0.114*
All	2.062	0.663	0.373	0.119	2.118	0.661	0.389	0.119	**1.746**	**0.622**	*0.332*	**0.109**	*1.764*	*0.629*	**0.290**	*0.112*

Table 3: Timing results KITTI dataset for the feature extraction and solver processing. Best values marked in **bold** and second best in *italic*. Time is measured in milliseconds.

Seq	**Extraction**								**GC-RANSAC**							
	PC/SIFT				AC				Nister (5PC)		Choi (2PC)		Guan (1AC)		Our (1SIFT)	
	Extr. time		Match. time		Extr. time		Match. time									
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
00	**58.98**	**58.75**	**4.30**	**3.58**	*1640.09*	*1597.54*	*6.42*	*5.77*	6.38	5.74	*6.01*	*5.40*	7.93	7.17	**5.76**	**5.17**
01	**53.51**	**52.98**	**1.61**	**0.90**	*844.71*	*636.66*	*2.85*	*1.78*	5.26	4.91	*4.87*	*4.60*	6.21	5.88	**4.62**	**4.34**
02	**56.63**	**56.70**	**7.35**	**6.90**	*2025.68*	*2112.82*	*9.98*	*9.52*	5.73	5.26	*5.08*	*4.71*	6.98	6.49	**4.99**	**4.59**
03	**57.01**	**57.03**	**7.13**	**6.72**	*2161.32*	*2222.02*	*9.51*	*9.08*	10.07	8.85	*8.89*	*7.85*	12.21	10.93	**8.75**	**7.74**
04	**54.39**	**54.27**	**4.37**	**4.17**	*1742.10*	*1745.43*	*7.14*	*6.50*	*6.25*	*5.95*	6.40	6.10	8.60	8.04	**6.05**	**5.94**
05	**56.27**	**56.27**	**4.77**	**4.16**	*1673.91*	*1632.84*	*7.24*	*6.65*	7.80	6.38	*7.26*	*6.17*	9.93	8.40	**7.20**	**5.99**
06	**54.66**	**53.87**	**3.59**	**3.11**	*1390.95*	*1379.29*	*5.45*	*5.33*	4.88	4.56	**4.72**	*4.55*	6.53	6.20	*4.73*	**4.39**
07	**54.85**	**54.73**	**4.50**	**4.14**	*1697.43*	*1698.24*	*6.61*	*6.01*	9.57	7.10	*9.20*	*6.79*	11.62	8.69	**8.86**	**6.66**
08	**55.36**	**55.41**	**6.07**	**5.55**	*1867.73*	*1912.30*	*8.41*	*7.88*	6.82	5.93	*6.18*	*5.32*	8.27	7.31	**5.83**	**5.14**
09	**54.88**	**54.69**	**5.41**	**4.18**	*1629.74*	*1604.78*	*7.44*	*6.40*	5.49	5.02	**4.89**	*4.52*	6.36	5.67	*4.91*	**4.46**
10	**55.31**	**55.13**	**5.28**	**4.51**	*1776.74*	*1720.86*	*7.95*	*7.02*	7.01	5.02	*6.66*	**4.75**	9.13	6.40	**6.53**	*4.80*
All	**57.22**	**57.36**	**5.08**	**4.19**	*1731.17*	*1701.15*	*7.33*	*6.51*	6.42	5.63	*5.90*	*5.18*	7.89	7.02	**5.72**	**5.03**

The results in accuracy are shown in Tab. 2 and the corresponding timings in Tab. 3. These are summarized in Tab. 4, where the trade-off between the total execution time and the accuracy for the different solvers are shown. In terms of accuracy, the 1 AC solver from Guan *et al*. achieves the best performance, and as such, it is selected as the baseline for comparison with the remaining solvers in Tab. 4; however, the solver requires extensive computational resources, both for the feature extraction (which was computed using GPU acceleration) and for the GC-RANSAC estimation. Our proposed method obtains similar accuracy as the baseline, improving both PC-based solvers, but with a substantial improvement of the execution times as it does not rely on affine features, nor GPU support. Indeed, the slight improvement of $0.02°$, $0.04°$ shown by Guan *et al*. does not justify the $25\times$ increase of computation, besides using GPU acceleration. In conclusion, these results demonstrate that approximating affine correspondences with SIFT features can lead to a solver with a superior trade-off between time and accuracy.

Table 4: Results on the KITTI dataset showing the trade-off between execution time and precision relative to the best performing method [30], based on the average over all sequences. Times are expressed in milliseconds and errors in degrees.

Solver	Tot. exec. time		Rel. exec. time		$\Delta\epsilon_{\mathbf{t}}$		$\Delta\epsilon_{\mathbf{R}}$	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Guan (1 AC)*	1746.39	1714.68	—	—	—	—	—	—
Nister (5PC)	68.72	67.18	25.4×	25.5×	−0.316	−0.041	−0.041	−0.010
Choi (2PC)	68.20	66.73	25.6×	25.7×	−0.372	−0.039	−0.057	−0.010
Our (1SIFT)	**68.02**	**66.58**	**25.7**×	**25.8**×	−0.018	−0.007	+0.042	−0.003

* Requires GPU acceleration.

Malaga Dataset Lastly, we consider the Malaga Dataset. Here, the ground-truth poses were obtained manually from the GPS readings, and for evaluation we used every 10th frame of the left camera, estimating the relative pose between consecutive images in a total of $\sim$ 9000 images. As in the previous experiment, we use the Stewenius solver for the larger-than-minimal sample in GC-RANSAC.

Fig. 6 shows the compared accuracy and GC-RANSAC processing time between the different methods, in the form of cumulative density functions, as in [31]. The curve approaching the left side of the plot means that a method is accurate or fast. The results show that all methods have a similar performance in accuracy terms, with the 1 AC solver displaying a slightly worse performance. Additionally, we noted a substantial speed reduction by the 1 AC solver ($\sim$ 10%), in comparison with the other solvers. Nevertheless, our proposal is again able to reach a solution comparable with the state-of-the-art in less time: our method has a 80% probability of reaching a solution in $\sim$ 27.5 ms, while others take $\sim$ 30 ms, meaning an improvement of at least 8%. The increase in processing time compared with the previous section is due to the higher number of features extracted in this dataset.

4 Discussion

Affine correspondences give the most information with three constraints per correspondence, followed by two correspondences for the proposed covariant features and only one for point-based solvers; however, failure to incorporate the trace constraint (7) is detrimental to the performance, as is illustrated in Fig. 3, Fig. 4, and Fig. 5. On the other hand, real-life scenarios are rarely perfectly planar. Applying nonlinear refinement should be considered good practice; however, previous authors have incorporated their solvers in a standard RANSAC. Using GC-RANSAC [6], we have demonstrated that the difference between the solvers in terms of performance is not very large when applying local optimization; hence, the processing time is the main useful metric. Comparing the results of a standard RANSAC implementation conducted in [30], the two-point method by Choi and Kim [14] has a median translational error reported between 2.8° and 6.4° for the KITTI sequences; however, embedded in GC-RANSAC this error is never above 0.89°, see Tab. 2. Moreover, this is superior to all other

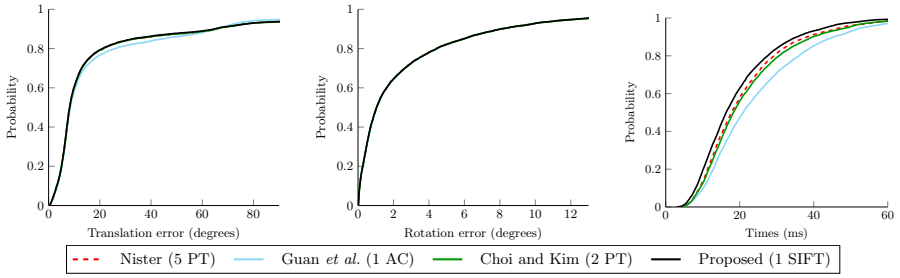

Fig. 6: *Malaga experiments.* Cumulative density functions deploying the performance of four different solvers in terms of angular error and GC-RANSAC processing time (excluding pre-processing time) for the Malaga dataset. A method being accurate (or fast) is equivalent to its curve being on the left side of the plot.

methods in the comparison of [30], including the affine-based solvers proposed by the authors. Far more important is the extraction time of state-of-the-art affine correspondences, such as AffNet [34] used in this paper. Even with GPU acceleration, it requires 25× more processing time (compared to SIFT measured on a CPU).

In this paper, we have not implemented histogram voting, as in [30]. The main reason for this is that any potential speed-up due to histogram voting cannot ameliorate the excessive processing time inflicted by the affine feature extraction and matching. Furthermore, the 1 SIFT method proposed in this paper may also reap the potential benefits from histogram voting, while also being faster, as demonstrated in Tab. 1. We do, however, include results in the supplementary material for those interested.

5 Conclusions

In this paper, we utilized a planar motion-compatible SIFT constraint to build a minimal 1 SIFT solver which is shown to be fast, stable, and performs on par with the current state-of-the-art. Furthermore, we demonstrated a significant speed-up compared to affine-based solvers and provided an in-depth comparison of point-based, SIFT-based, and affine correspondence-based estimation using a modern, robust estimation framework. The proposed solver was evaluated in controlled synthetic environments, where it was concluded that the trace constraint is important to the overall performance. On two well-established datasets for autonomous driving, we showed that the proposed method performs on par with the current state-of-the-art affine-based methods while accelerating the time by a factor of more than 25× in certain scenarios.

References

1. Barath, D., Molnar, J., Hajder, L.: Optimal surface normal from affine transformation. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (2015)
2. Barath, D.: Making sift features affine covariant. International Journal of Computer Vision (IJCV) **131** (2023)
3. Barath, D., Hajder, L.: Efficient recovery of essential matrix from two affine correspondences. IEEE Transactions on Image Processing **27**, 5328–5337 (2018)
4. Barath, D., Kukelova, Z.: Homography from two orientation- and scale-covariant features. In: International Conference on Computer Vision (ICCV) (2019)
5. Barath, D., Kukelova, Z.: Relative pose from sift features. In: European Conference on Computer Vision (2022)
6. Barath, D., Matas, J.: Graph-cut ransac. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
7. Barath, D., Mishkin, D., Cavalli, L., Sarlin, P.E., Hruby, P., Pollefeys, M.: Affineglue: Joint matching and robust estimation (2023)
8. Barath, D., Polic, M., Förstner, W., Sattler, T., Pajdla, T., Kukelova, Z.: Making Affine Correspondences Work in Camera Geometry Computation. In: European Conference on Computer Vision (ECCV) (2020)
9. Bentolila, J., Francos, J.M.: Conic epipolar constraints from affine correspondences. Computer Vision and Image Understanding **122**, 105–114 (2014)
10. Blanco-Claraco, J.L., Moreno-Duenas, F.A., González-Jiménez, J.: The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. The International Journal of Robotics Research (IJRR) **33**(2), 207–214 (2014)
11. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
12. Brown, M., Hartley, R.I., Nister, D.: Minimal solutions for panoramic stitching. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2007)
13. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics **32**(6), 1309–1332 (2016)
14. Choi, S., Kim, J.H.: Fast and reliable minimal relative pose estimation under planar motion. Image and Vision Computing **69**, 103–112 (2018)
15. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Pattern Recognition. pp. 236–243 (2003)
16. Codreanu, V., Dong, F., Liu, B., Roerdink, J.B., Williams, D., Yang, P., Yasar, B.: Gpu-asift: A fast fully affine-invariant feature extraction algorithm. In: International Conference on High Performance Computing & Simulation (HPCS). pp. 474–481 (2013)
17. Dibene, J.C., Min, Z., Dunn, E.: General planar motion from a pair of 3d correspondences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8060–8070 (October 2023)
18. Ding, Y., Barath, D., Kukelova, Z.: Homography-based egomotion estimation using gravity and sift features. In: Asian Conference on Computer Vision (ACCV) (2020)
19. Ding, Y., Barath, D., Kukelova, Z.: Minimal solutions for panoramic stitching given gravity prior. In: International Conference on Computer Vision (ICCV) (2021)
20. Ding, Y., Barath, D., Yang, J., Kong, H., Kukelova, Z.: Globally optimal relative pose estimation with gravity prior. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

21. Ding, Y., Barath, D., Yang, J., Kukelova, Z.: Relative pose from a calibrated and an uncalibrated smartphone image. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
22. Ding, Y., Yang, J., Ponce, J., Kong, H.: An efficient solution to the homography-based relative pose problem with a common reference direction. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)
23. Eichhardt, I., Chetverikov, D.: Affine correspondences between central cameras for rapid relative pose estimation. In: European Conference on Computer Vision (ECCV) (2018)
24. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
25. Fraundorfer, F., Tanskanen, P., Pollefeys, M.: A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In: European Conference on Computer Vision (ECCV) (2010)
26. Gal, I.G., Barath, D., Hajder, L.: Pose estimation for vehicle-mounted cameras via horizontal and vertical planes. In: International Conference on Robotics and Automation (ICRA) (2021)
27. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3354–3361 (June 2012)
28. Guan, B., Zhao, J., Barath, D., Fraundorfer, F.: Minimal cases for computing the generalized relative pose using affine correspondences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6068–6077 (October 2021)
29. Guan, B., Zhao, J., Barath, D., Fraundorfer, F.: Minimal solvers for relative pose estimation of multi-camera systems using affine correspondences. International Journal of Computer Vision (IJCV) **131** (2023)
30. Guan, B., Zhao, J., Li, Z., Sun, F., Fraundorfer, F.: Minimal solutions for relative pose with a single affine correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
31. Hajder, L., Barath, D.: Least-squares optimal relative planar motion for vehicle-mounted cameras. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 8644–8650 (2020)
32. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, England, UK, second edn. (2004)
33. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. Advances in neural information processing systems **30** (2017)
34. Mishkin, D., Radenović, F., Matas, J.: Repeatability is not enough: Learning affine regions via discriminability. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 287–304. Springer International Publishing, Cham (2018)
35. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences **2**, 438–469 (2009)
36. Nistér, D.: An Efficient Solution to the Five-Point Relative Pose Problem. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(6), 756–770 (6 2004)
37. Raposo, C., Barreto, J.P.: Theory and practice of structure-from-motion using affine correspondences. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

38. Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for pytorch. In: Winter Conference on Applications of Computer Vision (WACV) (2020), https://arxiv.org/pdf/1910.02190.pdf
39. Saurer, O., Vasseur, P., Boutteau, R., Demonceaux, C., Pollefeys, M., Fraundorfer, F.: Homography based egomotion estimation with a common direction. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(2), 327–341 (Feb 2017)
40. Saurer, O., Fraundorfer, F., Pollefeys, M.: Homography based visual odometry with known vertical direction and weak Manhattan world assumption. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vilamoura, Portugal (2012)
41. Scaramuzza, D.: 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. International Journal of Computer Vision (IJCV) **95**(1), 74–85 (October 2011)
42. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE robotics & automation magazine **18**(4), 80–92 (2011)
43. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
44. Stewart, C.V.: Robust parameter estimation in computer vision. SIAM Review **41**(3), 513–537 (1999)
45. Stewenius, H., Engels, C., Nister, D.: An efficient minimal solution for infinitesimal camera motion. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2007)
46. Valtonen Örnhag, M., Persson, P., Wadenbäck, M., Åström, K., Heyden, A.: Efficient real-time radial distortion correction for uavs. In: Winter Conference on Applications of Computer Vision (WACV) (2021)
47. Valtonen Örnhag, M., Persson, P., Wadenbäck, M., Åström, K., Heyden, A.: Minimal solvers for indoor uav positioning. In: International Conference on Pattern Recognition (ICPR) (2021)
48. Valtonen Örnhag, M., Persson, P., Wadenbäck, M., Åström, K., Heyden, A.: Trust your imu: Consequences of ignoring the imu drift. In: Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2022)
49. Valtonen Örnhag, M.: Fast non-minimal solvers for planar motion compatible homographies. In: International Conference on Pattern Recognition Applications and Methods (ICPRAM). pp. 40–51. Prague, Czech Republic (February 2019)
50. Valtonen Örnhag, M., Heyden, A.: Relative pose estimation in binocular vision for a planar scene using inter-image homographies. In: International Conference on Pattern Recognition Applications and Methods (ICPRAM). pp. 568–575. Funchal, Madeira, Portugal (January 2018)
51. Yu, Z., Guan, B., Liang, S., Li, Z., Ye, S., Yu, Q.: Globally optimal relative pose estimation using affine correspondences with known vertical direction. IEEE Transactions on Instrumentation and Measurement **72**, 1–12 (2023)
52. Zhao, C., Fan, B., Hu, J., Pan, Q., Xu, Z.: Homography-based camera pose estimation with known gravity direction for uav navigation. Science China Information Sciences **64** (2021)