Performance Evaluation and Statistical Analysis of Algorithms for Ego-Motion Estimation

Jan Erik Stellet[1], Christian Heigele[1], Florian Kuhnt[2], J. Marius Zöllner[2] and Dieter Schramm[3]

Abstract— **This contribution investigates algorithms for ego-motion estimation from environmental features. Various formulations for solving the underlying *procrustes problem* exist. It is analytically shown that in the 2-D case this can be performed more efficiently compared to common implementations based on matrix decompositions. Furthermore, analytic error propagation is performed to second order which reveals a multiplicative estimator bias. A novel bias-corrected solution is proposed and evaluated in Monte Carlo simulations. Propagation of the derived error model to a representation used in the recursive trajectory reconstruction is presented and verified.**

I. Introduction

Ego-motion estimation is an important topic both in robotics and advanced driver assistance (ADAS) applications. For an increasing number of upcoming automated driving functions, it is of paramount importance to accurately estimate the current vehicle position and driven trajectory.

Localisation approaches can be divided into global positioning against a known map and local, relative estimation with incremental mapping. One key advantage of employing relative position estimation is that it does not depend on external information sources such as GPS, which might be unavailable, e.g. in parking decks or tunnels.

The core problem of relative trajectory estimation is to reconstruct the vehicle's movements between two consecutive time steps. This can be accomplished based on inertial measurements (IMU), correspondences in static environmental features or a fusion of both. In this contribution it is studied how the relative motion can be estimated from two sets of corresponding feature points (*procrustes problem*). This underlying task is not dependent on a specific sensor technology.

The scope of this work is twofold: Firstly, different formulations of the procrustes solution are compared in terms of their processing times when implemented using standard libraries. This is especially relevant for the real-time requirements that are prevalent in the ADAS domain. Secondly, statistical properties of the estimate are studied. This has been recently demonstrated to first order for an algorithm in 3-D which is based on a singular value decomposition (SVD) [1]. In this work, second order approximations are calculated for the solution in 2-D which leads to simple closed-form expressions of the estimator's bias.

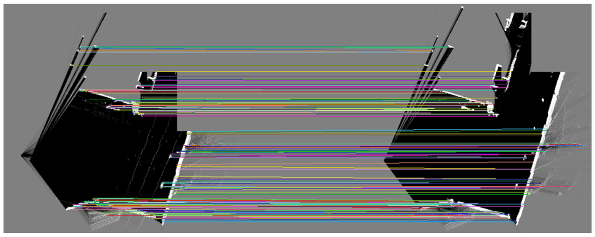


Fig. 1. The same scene at two different poses. Two environmental maps are shown that are built from stereo video measurements. From each map, a set of significant features is extracted and a corresponding match in the other data set is searched for. Invalid matches have already been removed.

As the eventual application, estimating a vehicle's trajectory over multiple time steps is considered. The recursive expressions for the trajectory reconstruction use a different representation of the incremental position updates. Propagation of the error model to this stage is analytically studied and verified with Monte Carlo simulations.

The rest of this work is organized as follows: Background, problem formulation and different versions of the solution are introduced in Sec. II. A computational analysis with both analytic and empirical evaluations is given in Sec. III. Results of statistical analysis of the rotation estimates and the position update are presented in Sec. IV and Sec. V. All findings are summarised in Sec. VI.

II. Background and previous results

A. Localisation based on correspondence sets

A frequently applied localisation strategy is to extract salient environment features from either a map or live measurements. These are compared and matched to a database or previous measurements, invalid matches are removed and corresponding features assigned. The corresponding sets are then used for incremental trajectory estimation. Essentially, the local transformation between two vehicle poses is estimated in order to support, correct or validate odometry, e.g. provided by a dead reckoning system.

Multiple feature descriptors have been proposed [2], these are either vision-based, e.g. SIFT [3], SURF [4] or ORB [5], geometric, e.g. line segments, or keypoints on occupancy grids [6] as in Fig. 1.

A critical point within such methods is to remove all invalid matches. This is often achieved by the use of a RANSAC algorithm, where a small number n of corresponding pairs is drawn from a tentative set and the best

[1] Jan Erik Stellet and Christian Heigele are with Robert Bosch GmbH, Corporate Research, Vehicle Safety and Assistance Systems, 71701 Schwieberdingen, Germany

[2] Florian Kuhnt and J. Marius Zöllner are with Research Center for Information Technology (FZI), 76131 Karlsruhe, Germany

[3] Dieter Schramm is with the Faculty of Engineering, University of Duisburg-Essen, 47048 Duisburg, Germany

matching transformation model for this subset is determined. The overall number of features within both maps that support this model is then identified. Since this procedure is typically executed for thousands of cycles, an efficient method for estimating the transformation model in each iteration has to be used. Especially embedded devices are severely limited in terms of computational performance and memory [6].

B. *Procrustes problem*

Let two registered sets of 2-D feature locations measured from a moving platform be denoted as $\{\mathbf{x}'_i\}_{i=1}^n$, $\{\mathbf{y}'_i\}_{i=1}^n$. As the features are assumed to be static, the relative location measurements systematically change due to the vehicle's movements. Transformation into the coordinate system of the current time step is described by translation $\mathbf{t}$ and by rotation angle ρ:

$$\mathbf{y}' = \mathbf{R}\mathbf{x}' + \mathbf{t} \tag{1}$$

where

$$\mathbf{R} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} \cos(\rho) & -\sin(\rho) \\ \sin(\rho) & \cos(\rho) \end{bmatrix} . \tag{2}$$

If at least $n \geq 2$ point correspondences are known, the ego-motion parameters $\mathbf{R}, \mathbf{t}$ can be retrieved. For the estimation, the sum of squared errors is used as a cost function:

$$J = \sum_{i=1}^{n} \|\mathbf{y}'_i - \mathbf{R}\mathbf{x}'_i - \mathbf{t}\|^2 . \tag{3}$$

In the case of isotropic, uncorrelated Gaussian noise, minimising this function yields an optimal estimate [7].

C. *Solution to the* 2*-D procrustes problem*

Various closed-form solutions to the *procrustes problem* (3) for the general 3-D case have been presented and include singular value decomposition [8], [9], polar decomposition [10] or quaternion representations [11], [12]. Apart from degenerate cases, these produce identical results [13].

In the domain of driver assistance, a planar world assumption is commonly made. For this two-dimensional case, the cost function (3) can be analytically minimised with respect to the entries of $\mathbf{R}$ and $\mathbf{t}$ [14]. Incorporating the non-linear constraints on the rotation parameters (2) gives the following Lagrangian problem formulation:

$$\min_{c,s,\mathbf{t}} \sum_{i=1}^{n} \|\mathbf{y}'_i - \mathbf{R}\mathbf{x}'_i - \mathbf{t}\|^2 + \Lambda(c^2 + s^2 - 1) . \tag{4}$$

The centroids of the sets of points are given as:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}'_i, \quad \bar{\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}'_i . \tag{5}$$

In the following, feature locations relative to the centroids are denoted as $\mathbf{x}_i = \mathbf{x}'_i - \bar{\mathbf{x}}$, $\mathbf{y}_i = \mathbf{y}'_i - \bar{\mathbf{y}}$ and $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \ldots & \mathbf{x}_n\end{bmatrix}$, $\mathbf{Y} = \begin{bmatrix}\mathbf{y}_1 & \ldots & \mathbf{y}_n\end{bmatrix}$.

Then, one obtains the following solution for (4) [14]:

$$c = \frac{f_1}{\sqrt{f_1^2 + f_2^2}}, \quad s = \frac{f_2}{\sqrt{f_1^2 + f_2^2}} \tag{6a}$$

$$\mathbf{t} = \bar{\mathbf{y}} - \mathbf{R}\bar{\mathbf{x}} \tag{6b}$$

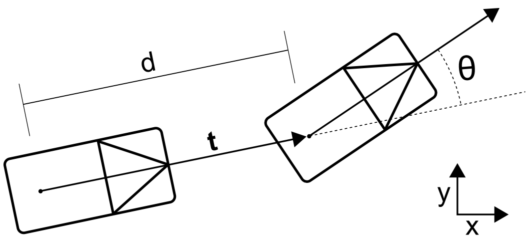


Fig. 2. Representation of incremental position updates.

with

$$f_1 = \sum_{i=1}^{n} \mathbf{x}_i^\top \mathbf{y}_i, \quad f_2 = \sum_{i=1}^{n} \mathbf{x}_i^\top \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{y}_i . \tag{7}$$

A recursive formulation is presented in [14].

D. *Trajectory reconstruction*

The previously described approach can be used to estimate relative motion parameters in 2-D between two consecutive time steps. In order to reconstruct a whole trajectory in a fixed coordinate system, a recursive formulation is used.

Denoting global position as $\mathbf{p}_{k-1} \in \mathbb{R}^2$ and orientation as ρ_{k-1} at time step $k-1$. A translation $\mathbf{t}_k$ of static world features is given relative to the vehicle's orientation and thus the global vehicle position updated as:

$$\mathbf{p}_k = \mathbf{p}_{k-1} - \begin{bmatrix} \cos(\rho_k) & -\sin(\rho_k) \\ \sin(\rho_k) & \cos(\rho_k) \end{bmatrix} \mathbf{t}_k \tag{8}$$

The updated vehicle orientation is calculated independently of the translation from the entries c_k, s_k in $\mathbf{R}_k$ as:

$$\rho_k = \rho_{k-1} - \arctan\left(\frac{s_k}{c_k}\right) . \tag{9}$$

A non-recursive form of this algorithm is given in [15]:

$$\mathbf{p}_k = \sum_{i=1}^{k} d_i \left[\cos\left(\sum_{j=1}^{i} \theta_j\right) \quad \sin\left(\sum_{j=1}^{i} \theta_j\right)\right]^\top , \tag{10}$$

where as illustrated in Fig. 2:

$$d_i = \|\mathbf{t}_i\| \tag{11}$$

$$\theta_i = \angle\left(-\mathbf{R}_i^\top \mathbf{t}_i\right) - \angle\left(-\mathbf{t}_{i-1}\right) . \tag{12}$$

This formulation is used in [15] for analysis of the cumulated errors in the estimated object trajectory $\mathbf{p}_k$. The study, however, is based on the assumption of Gaussian error distributions in d and θ. Within the same framework, these results will be extended in Sec. V of this work by studying error propagation from feature location measurements to d, θ.

III. Computational analysis

The explicit solution to the 2-D case from Sec. II-C incorporates only basic arithmetic operations. In contrast to methods that involve matrix decompositions, redundant calculations are avoided which is analytically shown in Sec. III-A. This motivates an empirical evaluation of processing times for various optimised implementations in Sec. III-B.

A. Analytic comparison to an SVD-based solution

A well-known solution to (3) can be obtained from an SVD of the matrix $\mathbf{A} := \mathbf{Y}^\top \mathbf{X}$. Denoting this decomposition as $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, one obtains the solution for $\mathbf{R}$ as [8]:

$$\mathbf{R} = \mathbf{U}\mathbf{C}\mathbf{V}^\top, \text{ with } \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & \det(\mathbf{U}\mathbf{V}^\top) \end{bmatrix} . \tag{13}$$

In the following it will be shown that while this algorithm gives the same solution as the closed-form expression (6) it includes redundant calculations.

From the definition of the SVD, $\mathbf{U}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^\top$ and $\mathbf{V}^\top$ the eigenvectors of $\mathbf{A}^\top\mathbf{A}$. The square roots of the corresponding eigenvalues determine the diagonal matrix $\boldsymbol{\Sigma}$. Because $\mathbf{U}, \mathbf{V}^\top$ are orthogonal matrices and all eigenvectors can be assumed normalised without loss of generality, these can be expressed as:

$$\mathbf{U} = \begin{bmatrix} c_\psi & -s_\psi \\ s_\psi & c_\psi \end{bmatrix}, \quad \mathbf{V}^\top = \begin{bmatrix} c_\phi & s_\phi \\ -s_\phi & c_\phi \end{bmatrix} . \tag{14}$$

Interpreting the matrix entries as triangular functions, the solution (13) is reformulated as:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\top = \begin{bmatrix} \cos(\psi - \phi) & -\sin(\psi - \phi) \\ \sin(\psi - \phi) & \cos(\psi - \phi) \end{bmatrix} . \tag{15}$$

It is immediately clear from this expression that calculating $\mathbf{R}$ by individual decomposition in $\mathbf{U}$ and $\mathbf{V}^\top$ induces redundant computations. Only the difference of the angles $\psi - \phi$ is required instead of the individual values, which constitute the full decomposition. It is shown in the appendix how this angle difference can be explicitly calculated leading to the same solution as (6).

B. Experimental comparison to standard implementations

Experiments have been performed to compare a plain C++ implementation[1] of the direct solution (6) against three different implementations of the SVD-based algorithm (13). These are built upon the SVD routines provided by the OpenCV 2.4.6 library, the LAPACK 3.5.0 library and an optimised implementation of the 2-D SVD from the `cgeom` library.[2]

Processing times for different numbers of samples are stated in Fig. 3. It can be observed that the plain implementation outperforms the SVD-based algorithms significantly. A linear dependence on the number of features n is indicated in both cases. This was expected, as the calculation of the matrix $\mathbf{A}$ for the SVD decomposition (13) and the factors f_1 and f_2 in (7) provide the same computational complexity. Further evaluations have been performed using MATLAB. Processing times of the direct implementation were in the order of 1×10^{-4} s instead of 1×10^{-6} s in the C++ case.

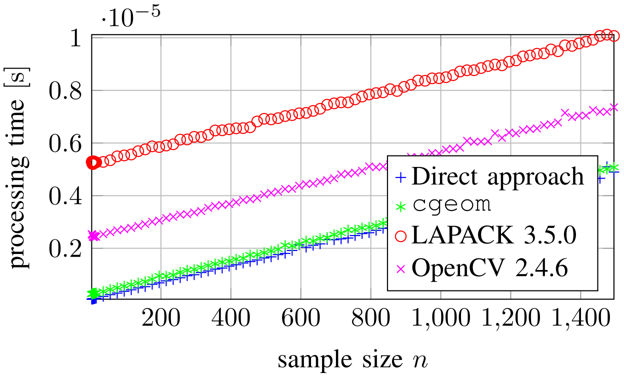


Fig. 3. Runtime analysis in C++ simulation with different algorithms. Time for estimation of the transformation parameters varying over the number of samples.

Some remarks on the practical relevance of these figures: The differences in the calculation speed seen in Fig. 3 might appear not too significant at first glance. However, if the transformation estimation is applied, e.g., in a RANSAC scheme, as described in Sec. II-A, its execution is repeated thousands of times. Given that a single run takes in the order of 1×10^{-6} s, the model-determining part of the RANSAC algorithm lasts in the order of 1×10^{-3} s. Therefore, an improvement in the runtime of one order,[3] can have a non-neglectable impact on the real-time capability of the overall procedure. For example, the algorithm used in [6] features an overall runtime of approximately 8 ms which would increase up to over 13 ms using OpenCV/LAPACK instead of the direct implementation.

In driver assistance applications, plain implementations are favoured over the use of external libraries for reasons of memory consumption, verification (ISO 26262) and legal aspects. Moreover, constraints in the type and the alignment of data structures can be avoided by not relying on external libraries. This is significant, e.g. for embedded devices where the available memory storage is really tight or special data types have to be used.

IV. Statistical Analysis

In this section it is analysed how zero mean stochastic errors $\Delta\mathbf{x}, \Delta\mathbf{y}$ added to the true feature positions affect the estimation results. It is assumed that these errors are statistically independent. Emphasis is put on the difference between the true value of the estimated rotation and the expectation of its estimate (bias). Non-zero mean measurement errors or incorrect approximations of skewed error distributions [16] are common cause for biased estimates. A second cause is that even for symmetric, zero mean error densities, non-linear estimation algorithms produce biased estimates [17]. In the following, this inherent property of the relative localisation algorithm (6) will be studied.

[1] Optimisations, possibly with consideration of special processor instructions could improve the computation speed further.

[2] `https://github.com/victorliu/Cgeom`

[3] For sample numbers $n \leq 10$ the difference between the optimized 2×2 SVD and the direct solution is hard to spot in Fig. 3 but it takes approx. 5×10^{-8} s for the direct approach and 3×10^{-7} s for the SVD.

A. Anisotropic Gaussian noise

Firstly, the pertubations in f_1, f_2 are calculated from (7):

$$\Delta f_1 = \sum_{i=1}^{n} \mathbf{x}_i^\top \Delta\mathbf{y}_i + \mathbf{y}_i^\top \Delta\mathbf{x}_i + \Delta\mathbf{x}_i^\top \Delta\mathbf{y}_i \tag{16a}$$

$$\begin{aligned}\Delta f_2 = \sum_{i=1}^{n} \mathbf{x}_i^\top \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix} \Delta\mathbf{y}_i + \mathbf{y}_i^\top \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}^\top \Delta\mathbf{x}_i \\ + \Delta\mathbf{x}_i^\top \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix} \Delta\mathbf{y}_i \;.\end{aligned} \tag{16b}$$

A first-order approximation is used in [1] by neglecting the last term in these expressions.

Given a stochastic error model for $\Delta\mathbf{x}, \Delta\mathbf{y}$, the above expressions allow to calculate the error distribution of $\Delta\mathbf{f}$. In the case of Gaussian zero mean errors $\Delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_x)$, $\Delta\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_y)$ a Gaussian distribution $\Delta\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_f)$ is obtained.

Secondly, it is analysed how the Gaussian errors in $\Delta\mathbf{f}$ are mapped to the estimated values c, s. The rotation estimation formulae (6a) describe a non-linear function of $\mathbf{f}$. While previous research [1] has investigated this problem to first order, a second-order series expansion will be employed here. This allows to analyse the estimator's bias.

Taylor expansion of (6a) to second order is used to calculate pertubations $\Delta c, \Delta s$ from the true values c_0, s_0:

$$\Delta c := c - c_0 \approx \mathbf{g}_c^\top \Delta\mathbf{f} + \frac{1}{2} \Delta\mathbf{f}^\top \mathbf{H}_c \Delta\mathbf{f} \tag{17a}$$

$$\Delta s := s - s_0 \approx \mathbf{g}_s^\top \Delta\mathbf{f} + \frac{1}{2} \Delta\mathbf{f}^\top \mathbf{H}_s \Delta\mathbf{f} \tag{17b}$$

with

$$\mathbf{g}_c^\top = \frac{1}{\left(f_1^2 + f_2^2\right)^{\frac{3}{2}}} \begin{bmatrix} f_2^2 & -f_1 f_2 \end{bmatrix} \tag{18a}$$

$$\mathbf{g}_s^\top = \frac{1}{\left(f_1^2 + f_2^2\right)^{\frac{3}{2}}} \begin{bmatrix} -f_1 f_2 & f_1^2 \end{bmatrix} \tag{18b}$$

and

$$\mathbf{H}_c = \frac{1}{\left(f_1^2 + f_2^2\right)^{\frac{5}{2}}} \begin{bmatrix} -3 f_1 f_2^2 & 2 f_1^2 f_2 - f_2^3 \\ 2 f_1^2 f_2 - f_2^3 & 2 f_1 f_2^2 - f_1^3 \end{bmatrix} \tag{19a}$$

$$\mathbf{H}_s = \frac{1}{\left(f_1^2 + f_2^2\right)^{\frac{5}{2}}} \begin{bmatrix} 2 f_1^2 f_2 - f_2^3 & 2 f_1 f_2^2 - f_1^3 \\ 2 f_1 f_2^2 - f_1^3 & -3 f_1^2 f_2 \end{bmatrix} \;. \tag{19b}$$

Then, the expectation of $\Delta c, \Delta s$ is calculated from (17):

$$\mathbb{E}\left[\Delta c\right] = \frac{1}{2} \operatorname{tr}\left(\mathbf{H}_c \mathbf{C}_f\right), \quad \mathbb{E}\left[\Delta s\right] = \frac{1}{2} \operatorname{tr}\left(\mathbf{H}_s \mathbf{C}_f\right) \;. \tag{20}$$

This expression shows that even in the case of zero mean input errors, the estimator's output is biased. Secondly, the variance of $\begin{bmatrix}\Delta c & \Delta s\end{bmatrix}^\top$ is obtained to first order:

$$\operatorname{Var}\left(\begin{bmatrix}\Delta c \\ \Delta s\end{bmatrix}\right) = \begin{bmatrix}\mathbf{g}_c^\top \\ \mathbf{g}_s^\top\end{bmatrix} \mathbf{C}_f \begin{bmatrix}\mathbf{g}_c & \mathbf{g}_s\end{bmatrix} \;. \tag{21}$$

B. Isotropic Gaussian noise

In the special case of uncorrelated isotropic noise with $\mathbf{C}_x = \sigma_x^2 \mathbf{I}$, $\mathbf{C}_y = \sigma_y^2 \mathbf{I}$, the previously derived expressions can be simplified.

Firstly, it is found using (16) that $\mathbf{C}_f = \sigma_f^2 \mathbf{I}$ with

$$\sigma_f^2 = \sigma_x^2 \operatorname{tr}\left(\mathbf{Y}\mathbf{Y}^\top\right) + \sigma_y^2 \operatorname{tr}\left(\mathbf{X}\mathbf{X}^\top\right) + 2n\sigma_x^2 \sigma_y^2 \;. \tag{22}$$

Then, (20)-(21) result in:

$$\mathbb{E}\left[\begin{bmatrix}\Delta c \\ \Delta s\end{bmatrix}\right] = -\frac{1}{2} \frac{1}{f_1^2 + f_2^2} \sigma_f^2 \begin{bmatrix} c_0 \\ s_0 \end{bmatrix} \tag{23a}$$

$$\operatorname{Var}\left(\begin{bmatrix}\Delta c \\ \Delta s\end{bmatrix}\right) = \frac{1}{f_1^2 + f_2^2} \sigma_f^2 \begin{bmatrix} s_0^2 & -c_0 s_0 \\ -c_0 s_0 & c_0^2 \end{bmatrix} \;. \tag{23b}$$

From (23a) follows that the estimator's bias is proportional to the absolute values with the relative estimation bias

$$\lambda = \frac{1}{2} \frac{1}{f_1^2 + f_2^2} \sigma_f^2 \;. \tag{24}$$

Error propagation to the estimated translation $\mathbf{t}$ from (6b) is straight-forward:

$$\mathbb{E}\left[\Delta\mathbf{t}\right] = -\lambda \begin{bmatrix} \bar{x}_1 & -\bar{x}_2 \\ \bar{x}_2 & \bar{x}_1 \end{bmatrix} \begin{bmatrix} c_0 \\ s_0 \end{bmatrix}, \quad \operatorname{Var}\left(\Delta\mathbf{t}\right) = \mathbf{C}_{\Delta\mathbf{t}} \tag{25}$$

$$\begin{aligned} \text{with} \quad C_{\Delta\mathbf{t},11} &= 2\lambda \left(c_0 \bar{x}_2 + s_0 \bar{x}_1\right)^2 + n^{-1}\left(\sigma_x^2 + \sigma_y^2\right) \\ C_{\Delta\mathbf{t},12} &= -2\lambda \left(c_0 \bar{x}_2 + s_0 \bar{x}_1\right)\left(c_0 \bar{x}_1 - s_0 \bar{x}_2\right) \\ C_{\Delta\mathbf{t},21} &= C_{\Delta\mathbf{t},12} \\ C_{\Delta\mathbf{t},22} &= 2\lambda \left(c_0 \bar{x}_1 - s_0 \bar{x}_2\right)^2 + n^{-1}\left(\sigma_x^2 + \sigma_y^2\right) \;. \end{aligned}$$

C. Bias-corrected algorithm

Based on the previous findings, a debiased estimator formula will be proposed. As there is in fact a multiplicative disturbance, a correction factor is added to (6a). As the correction factor is identical for both estimates, orthogonality of $\mathbf{R}$ is preserved:

$$c_{\text{new}} = \frac{f_1}{\sqrt{f_1^2 + f_2^2}} \frac{1}{1 - \lambda} \approx \frac{f_1}{\sqrt{f_1^2 + f_2^2}} \left(1 + \lambda\right) \tag{26a}$$

$$s_{\text{new}} = \frac{f_2}{\sqrt{f_1^2 + f_2^2}} \frac{1}{1 - \lambda} \approx \frac{f_2}{\sqrt{f_1^2 + f_2^2}} \left(1 + \lambda\right) \;. \tag{26b}$$

It has to be noted that the bias has been analytically derived relative to the true values c_0, s_0 which are not known in practice. Therefore the estimated values are used in the correction (expectation of bias).

D. Monte Carlo experiment

In order to verify the bias estimation calculated in the previous section, Monte Carlo simulations are used. Feature points $\left\{\mathbf{x}'_{i,0}\right\}_{i=1}^{n}$ are chosen from a uniform distribution in $[-1, 1] \times [-1, 1]$. Concentrating on the rotation estimation alone, idealised locations of $\left\{\mathbf{y}'_{i,0}\right\}_{i=1}^{n}$ are obtained by rotation with angle ρ according to (1). Simulated measurements $\mathbf{x}'_i, \mathbf{y}'_i$ are generated by adding uncorrelated, isotropic zero mean Gaussian errors with $\sigma = 0.2$ to both sets.

First, the rotation angle ρ is varied for a fixed set of $n = 10$ feature points. With $N = 10^6$ realisations of measurement noise each, the mean estimation bias for the

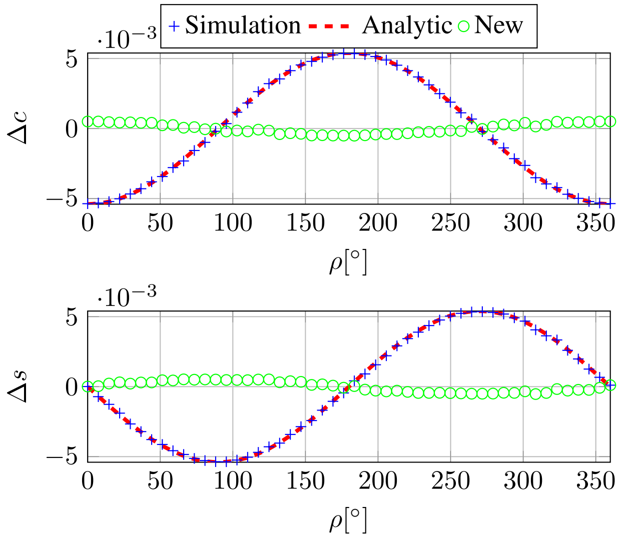


Fig. 4. Monte Carlo simulation results for $N = 10^6$ iterations using $n = 10$ uniformly distributed feature points $\mathbf{x}'_i \in [-1,1] \times [-1,1]$. Measurement noise is $\sigma = 0.2$. Estimation bias in rotation parameters c, s according to (6) and the bias-corrected $c_{\text{new}}, s_{\text{new}}$ from (26) is plotted over rotation angle ρ. The analytic expectation (23a) matches the empirical simulation results and the proposed novel solution yields almost bias-free estimates. The bias-correction is calculated using the uncertain measured quantities only instead of the exact c_0, s_0.

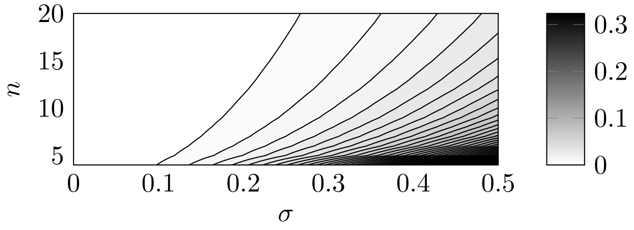


Fig. 5. Mean relative estimation bias λ according to (24) for sets of uniformly distributed feature points, rotation angle $\rho = 45°$ and isotropic measurement noise with standard deviation σ. The relative bias grows for increased measurement noise and smaller sample size n.

standard algorithm (6) is calculated and compared to the analytic expectation (23a). Results in Fig. 4 show that the bias in Δc, Δs is proportional to the absolute values $\cos(\rho)$, $\sin(\rho)$. A good correspondence between analytic and simulated values is obtained. Moreover, the mean simulation bias is evaluated for the novel algorithm (26) and almost bias-free estimates are obtained.

Secondly, the relative estimation bias λ from (24) is evaluated for different configurations of sample size n and measurement noise σ. In order to eliminate the influence of the distribution of feature locations, 10^4 realisations of uniformly distributed sets $\left\{\mathbf{x}'_{i,0}\right\}_{i=1}^{n}$ are generated.

Figure 5 visualises the mean relative bias for each configuration. It can be noted that especially for small sample sizes, the bias easily reaches values of 5% or more. While small systematic deviations might be acceptable for one-time evaluation, the error will typically be integrated over time in common applications. That is why the novel formula (26) is proposed to achieve bias-free estimates.

V. Error propagation in trajectory reconstruction

The trajectory reconstruction process from Sec. II-D uses a representation $(\mathbf{R}, \mathbf{t}) \to (d, \theta)$ for the incremental position updates. Given the previously derived error model for estimates $\mathbf{R}, \mathbf{t}$, error propagation to d and θ from (11)-(12) will be studied.

A. Length d of incremental update

First, error propagation to the norm d of the translation vector $\mathbf{t}$ whose solution is given by (6b) is analysed:

$$d = \|\mathbf{t}\| = \sqrt{t_x^2 + t_y^2}\,. \tag{27}$$

This will be performed by second order Taylor series expansion of (27) with respect to the estimated translation parameters t_x, t_y. The linear part of the approximation is used together with $\mathbf{C}_{\Delta\mathbf{t}}$ from (25) to calculate the variance σ_d^2. One obtains with $\boldsymbol{\tau} := \mathbf{t}/\|\mathbf{t}\|$ as the normalised translation vector:

$$\operatorname{Var}(d) = 2\lambda \left(\boldsymbol{\tau}^\top \mathbf{R} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{x}}\right)^2 + \frac{\sigma_x^2 + \sigma_y^2}{n}\,. \tag{28}$$

Second order expansion yields the expected value of Δd:

$$\mathbb{E}\left[\Delta d\right] = \frac{1}{d}\left(\lambda \left(\boldsymbol{\tau}^\top \mathbf{R}\bar{\mathbf{x}}\right)^2 + \frac{\sigma_x^2 + \sigma_y^2}{2n}\right)\,. \tag{29}$$

B. Orientation θ of incremental update

The calculation of θ involves estimates of the rotation matrix $\mathbf{R}_i$ and translation vectors $\mathbf{t}_i, \mathbf{t}_{i-1}$ from two different time-steps. As measurement errors are assumed independent over time, the two summands $\theta_1 := \angle - \mathbf{R}_i^\top \mathbf{t}_i$ and $\theta_2 := \angle - \mathbf{t}_{i-1}$ in (12) will be treated separately.

First, Gaussian error propagation with $\mathbf{C}_{\Delta\mathbf{t}}$ from (25) yields for the variance in $\Delta\theta_2$:

$$\operatorname{Var}(\Delta\theta_2) = \frac{1}{d^2}\left(2\lambda \left(\boldsymbol{\tau}^\top \bar{\mathbf{y}}\right)^2 + \frac{\sigma_x^2 + \sigma_y^2}{n}\right)\,. \tag{30}$$

A similar reasoning is used to calculate the variance in $\Delta\theta_1$. Here, a virtual measurement $\mathbf{t}' := -\mathbf{R}^\top \mathbf{t}$ is defined which allows to employ the same partial derivatives as in (30). Covariance $\mathbf{C}_{\Delta\mathbf{t}'}$ is derived using (23b). This results in:

$$\operatorname{Var}(\Delta\theta_1) = \frac{1}{d^2}\left(2\lambda \left(\boldsymbol{\tau}^\top \mathbf{R}\bar{\mathbf{x}}\right)^2 + \frac{\sigma_x^2 + \sigma_y^2}{n}\right)\,. \tag{31}$$

Finally, one obtains $\operatorname{Var}(\Delta\theta) = \operatorname{Var}(\Delta\theta_1) + \operatorname{Var}(\Delta\theta_2)$ with expressions (30)-(31) evaluated for the respective time steps.

With a second order expansion, the bias is calculated as:

$$\mathbb{E}\left[\Delta\theta_1\right] = \frac{2\lambda}{d^2}\left(\boldsymbol{\tau}^\top \mathbf{R}\bar{\mathbf{x}}\right)\left(\boldsymbol{\tau}^\top \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{y}}\right) \tag{32}$$

$$\mathbb{E}\left[\Delta\theta_2\right] = \frac{2\lambda}{d^2}\left(\boldsymbol{\tau}^\top \bar{\mathbf{y}}\right)\left(\boldsymbol{\tau}^\top \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{y}}\right)\,. \tag{33}$$

C. Monte Carlo experiment

The previously derived expressions for bias and variance are verified in Monte Carlo simulations. A set of $n = 10$ uniformly distributed features $\mathbf{x}_0' \in [0\,\mathrm{m},\ 20\,\mathrm{m}]^2$ is generated. Because the last terms in (28)-(29) depend on the relation between $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ and the ego-rotation ρ, different translations $\mathbf{t}$ with $\theta = \angle\mathbf{t}$ and $\|\mathbf{t}\| = 1\,\mathrm{m}$ are employed while $\rho = 20°$ is fixed. Thus, the virtual measurements $\mathbf{y}_0' = \mathbf{R}\mathbf{x}_0' + \mathbf{t}$ are obtained. Isotropic zero mean Gaussian errors with $\sigma = 0.2\,\mathrm{m}$ are added to both sets.

Simulation results are visualised in Fig. 6. Both bias and variance in Δd exhibit a fixed offset and an alternating component which depends on the orientation of $\mathbf{t}$. The analytic expressions closely match the simulation results. Here, the bias in Δd reaches up to 2% of the absolute value. This is because taking the norm (27) of zero mean Gaussian random variables $\Delta t_x, \Delta t_y$ produces a skewed distribution with non-zero mean in Δd. Simulations with varying ρ yield similar results with the same mean variance over θ.

For the derived error model in $\Delta\theta$, a good correspondence to the simulation results in Fig. 6(b) is observed as well. As the simulation considers only a single step, $\Delta\theta_1$ and $\Delta\theta_2$ are evaluated individually.

D. Effect of bias in Δd on cumulative trajectory error

The analytic expression (29) for the bias in Δd and the exemplary simulation results in Fig. 6(a) indicate a non-negligible drift in the length of each incremental position update. As the values d_i are accumulated in the position reconstruction (10) one intuitively expects a constant drift over time in the estimated position $\mathbf{p}_k$.

However, as will be briefly shown in the following, the bias in Δd is in fact compensated for by the angular noise process $\Delta\theta$. Therefore, correcting the bias in d alone would reduce the achieved accuracy. In order to illustrate this, the incremental update in the x-component for the first time step is considered as a special case of (10):

$$p_x = d\cos(\theta) = (d_0 + \Delta d)\cos(\theta_0 + \Delta\theta) \ . \tag{34}$$

For simplified expressions, assume $\theta_0 = 0$. Then, under the assumption of independence between $\Delta d, \Delta\theta$:

$$\begin{aligned}\mathbb{E}[p_x - p_{x,0}] &= (d_0 + \mathbb{E}[\Delta d])\,\mathbb{E}[\cos(\Delta\theta)] - d_0 \\ &= (d_0 + \mathbb{E}[\Delta d])\exp\left(-0.5\sigma_{\Delta\theta}^2\right) - d_0 \ .\end{aligned} \tag{35}$$

Inserting $\mathbb{E}[\Delta d]$ from (29), $\sigma_{\Delta\theta}^2$ from (33) and using the approximation $\exp\left(-0.5\sigma_{\Delta\theta}^2\right) \approx 1 - 0.5\sigma_{\Delta\theta}^2$ gives:

$$\mathbb{E}[p_x - p_{x,0}] \approx -\frac{\mu_{\Delta d}^2}{d} - \lambda\boldsymbol{\tau}^\top\bar{\mathbf{y}}\left(\frac{\mu_{\Delta d}}{d} + 1\right) - \lambda\boldsymbol{\tau}^\top\mathbf{R}\bar{\mathbf{x}} \ . \tag{36}$$

For typical values of the sensor and measurement characteristics, one can verify $\frac{\mu_{\Delta d}}{d} \ll 1$ and finally that the whole expression is smaller than $\mu_{\Delta d}$ by approximately one order of magnitude. Hence, the bias in Δd is compensated for by the induced bias in $\cos(\Delta\theta)$. It needs to be remarked that a long term drift of the estimated trajectory is expected due to the noise in $\Delta\theta$ which is studied in [15].

VI. Conclusion

In this work, low-level algorithms for solving the problem of relative position estimation have been studied. Focussing on the two-dimensional case, with its predominant applications in the driver assistance domain, computational and statistical analyses have been performed.

A significant speed-up is revealed when comparing a more efficient solution to algorithms based on matrix decompositions. It is analytically shown that inherently redundant computations in algorithms based on singular value decompositions can be avoided.

For the statistical analysis, error propagation in closed form with second order expansion has been facilitated by the direct expressions. In order not to be restricted to a specific sensor technology, a generic measurement model has been assumed. Expected values of the estimator bias have been derived and a new bias-corrected solution has been proposed. Monte Carlo simulations show that the novel formula yields estimates with bias reduced by one order of magnitude.

The error model for the single step position update is propagated to a representation used for recursive trajectory reconstruction. Derived expressions for bias and variance consist of fixed offset values and periodic components. This new error model allows to extend the work [15] on cumulative position drift in the reconstructed trajectory.

Experimental application of the provided scheme in a vehicle localisation task using a grid-based environment representation is demonstrated in [6].

Appendix

Starting with the definitions of $\mathbf{U}$ and $\mathbf{V}^\top$:

$$\mathbf{L} := \mathbf{A}\mathbf{A}^\top = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\mathbf{U}^\top \tag{37a}$$

$$\mathbf{M} := \mathbf{A}^\top\mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}\mathbf{V}^\top \ . \tag{37b}$$

The entries of the diagonal matrix $\boldsymbol{\Sigma}$ are the eigenvalues of $\mathbf{A}\mathbf{A}^\top$ denoted as σ_1, σ_2. Then, the following equations can be obtained from (37):

$$L_{11} - L_{22} = \cos(2\psi)\left(\sigma_1^2 - \sigma_2^2\right) \tag{38a}$$

$$M_{11} - M_{22} = \cos(2\phi)\left(\sigma_1^2 - \sigma_2^2\right) \tag{38b}$$

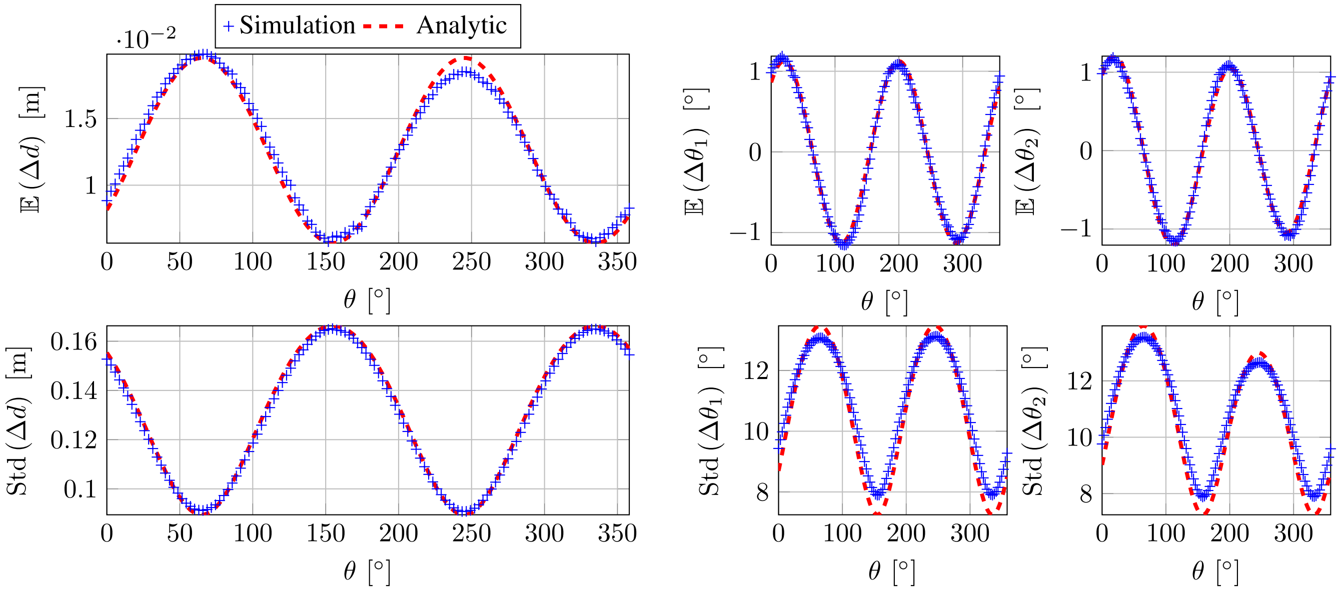
$$L_{12} + L_{21} = \sin(2\psi)\left(\sigma_1^2 - \sigma_2^2\right) \tag{38c}$$

$$M_{12} + M_{21} = \sin(2\phi)\left(\sigma_1^2 - \sigma_2^2\right) \ . \tag{38d}$$

Hence, the angles ψ and ϕ and thus the decomposition in $\mathbf{U}, \mathbf{V}^\top$ according to (14) can be calculated:

$$g := \tan(2\psi) = \frac{L_{12} + L_{21}}{L_{11} - L_{22}},\ h := \tan(2\phi) = \frac{M_{12} + M_{21}}{M_{11} - M_{22}} \ .$$

Note that it is not necessary to explicitly calculate the eigenvalues σ_1, σ_2, i.e, $\boldsymbol{\Sigma}$ from the full decomposition. Applying



(a) Bias and standard deviation in d compared to (28)-(29).

(b) Bias and standard deviation in $\Delta\theta_1, \Delta\theta_2$ compared to (30)-(33).

Fig. 6. Monte Carlo simulation results ($N = 10^6$ iterations) of the incremental position updates. A set of $n = 10$ uniformly distributed features $\mathbf{x}_0' \in [0\,\mathrm{m},\ 20\,\mathrm{m}]^2$ and varying direction of the translation $\theta = \angle\mathbf{t}$ with $\|\mathbf{t}\| = 1\,\mathrm{m}$ are used. Rotation is $\rho = 20°$ and measurement noise $\sigma = 0.2\,\mathrm{m}$.

trigonometric relationships, one obtains:

$$\cos(\psi - \phi) = \frac{1}{\sqrt{2}}\left(1 + 1\Big/\sqrt{1 + \left(\frac{g-h}{1+gh}\right)^2}\right)^{\frac{1}{2}} \tag{39a}$$

$$\sin(\psi - \phi) = \frac{1}{\sqrt{2}}\left(1 - 1\Big/\sqrt{1 + \left(\frac{g-h}{1+gh}\right)^2}\right)^{\frac{1}{2}}. \tag{39b}$$

Replacing g and h and inserting the elements of $\mathbf{L}$ and $\mathbf{M}$ from (37) gives

$$\sqrt{1 + \left(\frac{g-h}{1+gh}\right)^2} = \frac{(A_{11} + A_{22})^2 + (A_{12} - A_{21})^2}{(A_{11} + A_{22})^2 - (A_{12} - A_{21})^2} \tag{40}$$

and the entries of $\mathbf{R}$ can be finally expressed as:

$$\cos(\psi - \phi) = \frac{A_{11} + A_{22}}{\sqrt{(A_{11} + A_{22})^2 + (A_{12} - A_{21})^2}} \tag{41a}$$

$$\sin(\psi - \phi) = \frac{A_{12} - A_{21}}{\sqrt{(A_{11} + A_{22})^2 + (A_{12} - A_{21})^2}}. \tag{41b}$$

From (7) it is clear that (41) present the same result as (6).

References

[1] L. Dorst, "First order error propagation of the procrustes method for 3d attitude estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 2, pp. 221–229, 2005.

[2] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[3] D. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, Proceedings of the 7th IEEE International Conference on*, vol. 2, 1999, pp. 1150–1157.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 13th IEEE International Conference on*, Nov 2011, pp. 2564–2571.

[6] C. Heigele, H. Mielenz, J. Heckel, and D. Schramm, "Fast and accurate localization in unstructured environment based on shape context keypoints," in *Information Fusion (FUSION), 17th International Conference on Information Fusion*, 2014.

[7] K. Kanatani, "Analysis of 3-d rotation fitting," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 5, pp. 543–549, 1994.

[8] K. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[9] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, no. 4, pp. 376–380, 1991.

[10] B. K. P. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America*, vol. 5, no. 7, pp. 1127–1135, 1988.

[11] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987.

[12] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-d location parameters using dual number quaternions," *CVGIP: Image Underst.*, vol. 54, no. 3, pp. 358–367, 1991.

[13] D. Eggert, A. Lorusso, and R. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[14] C. Hermes, "Aktionserkennung und -prädiktion mittels trajektorienklassifikation," Ph.D. dissertation, Bielefeld University, 2012.

[15] F. Zhang, C. Simon, G. Chen, C. Buckl, and A. Knoll, "Cumulative error estimation from noisy relative measurements," in *Intelligent Transportation Systems - (ITSC), 16th International IEEE Conference on*, 2013, pp. 1422 – 1429.

[16] G. Dubbelman and F. Groen, "Bias reduction for stereo based motion estimation with applications to large scale visual odometry," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2009, pp. 2222–2229.

[17] Y. Ji, C. Yu, B. D. O. Anderson, and S. Drake, "A generic bias-correction method with application to scan-based localization," in *Control and Automation (ICCA), 9th IEEE International Conference on*, 2011, pp. 24–29.