Error Propagation in Monocular Navigation for Z_∞ Compared to Eightpoint Algorithm

Elmar Mair[1,2], Michael Suppa[1], and Darius Burschka[2]

Abstract— **Efficient visual pose estimation plays an important role for a variety of applications. To improve the quality, the measurements from different sensors can be fused. However, a reliable fusion requires the knowledge of the uncertainty of each estimate. In this work, we provide an error analysis for the Z_∞ algorithm. Furthermore, we extend the existing first-order error propagation for the 8-point algorithm to allow for feature normalization, as proposed by Hartley or Mühlich, and the rotation matrix based decomposition. Both methods are efficient visual odometry techniques which allow high frame-rates and, thus, dynamic motions in unbounded workspaces. Finally, we provide experiments which validate the accuracy of the error propagation and which enable a brief comparison, showing that the Z_∞ significantly outperforms the 8-point algorithm. We also discuss the influence of the number of features, the aperture angle, and the image resolution on the accuracy of the pose estimation.**

I. Motivation

There is a strong trend towards the use of cameras for pose estimation on mobile robots. They have low power consumption, large measurement fields, and are available in lightweight and compact housings. However, due to the large data volume, image processing is a computationally expensive task, which, in general, requires significant resources on robotic platforms. Hence, for monocular pose estimation on resource limited systems efficient algorithms have to be used. Depending on the environment, the camera parameters and the image quality, the uncertainty of any image based motion estimation can vary significantly. To allow for a proper post-processing of these measurements, like the fusion with other data, an accuracy uncertainty estimate for each measurement is crucial. The only information available to estimate the expected error are the feature correspondences which are used as input for the respective algorithm, the camera parameters, the expected tracking accuracy in the image, and the motion estimation result.

Several solutions already exist to estimate the camera motion from a monocular image stream. Especially in the last decade novel solutions, like MonoSLAM [1], PTAM [6] or DTAM [10], have been presented, which allow for an accurate and reliable motion estimation. However, they are quite memory and processing intensive and are restricted to a limited workspace. Hand-held systems, wearable devices, micro aerial or ground robots, they all have in common to provide only very limited processing resources, whereas requiring a pose estimation within an unbound workspace. Furthermore, the high dynamic motions in such applications demand an image processing at high frame-rates.

In this work, we derive a first-order error propagation for two highly efficient visual pose estimation algorithms, the Z_∞ and the normalized 8-point algorithm. We introduce a first-order error propagation for the rotation and the translation estimation of the Z_∞ algorithm [8], which allows for a fast motion estimation due to its closed-form solutions. There is some work in literature analysing the perturbation of the Fundamental matrix computation [15], [11], [12]. These solutions do not consider the motion estimation from an Essential matrix and are computationally complex or limited to a specific number of features. A straight forward first-order error propagation for the original 8-point algorithm including the motion estimation based on the Essential matrix is shown in [14]. We will extend this work taking into consideration the effect of feature normalization as proposed by Hartley [4] or Mühlich [9] which improves the estimation result significantly. We also propagate the uncertainty for the rotation matrix based decomposition, which is computationally more robust than the quaternion estimation as used in [14]. Additionally, we show how to compute the axis and the magnitude of the error rotation which allows for a geometrical interpretation of the error. The contribution of this work is a set of equations, including their derivations, which can be directly applied to any application which requires an efficient image-based pose estimation and the corresponding uncertainty measure. By comparing the results, we also show that the Z_∞ greatly outperforms the 8-point algorithm. The experiments reveal also some insights how the estimation accuracy correlates with the aperture angle, the number of features and the tracking accuracy.

In the following, we briefly sketch the Z_∞ and the 8-point algorithm to ease the understanding for the successive first-order uncertainty derivations in Sections III and IV. In Section V, we evaluate the accuracy of the error propagation and briefly compare both motion estimation techniques.

II. Mathematical Framework

In the following, we assume intrinsically calibrated cameras and undistorted image features. Let p denote the image coordinates of a specific feature corresponding to the 3D-point $P = \begin{pmatrix} x_P & y_P & z_P \end{pmatrix}^T$. The respective projection onto a camera plane with focal length f is referred to as $\tilde{r} = \begin{pmatrix} x_{\tilde{r}} & y_{\tilde{r}} & f \end{pmatrix}^T$, the projection onto the unit focal camera

This work was supported by the German Aerospace Center (DLR).

[1]These authors are with the Department of Robotics and Mechatronics at the German Aerospace Center (DLR), 82234 Wessling, Germany {elmar.mair|michael.suppa} at dlr.de

[2]These authors are with the Faculty of Computer Science at the Technische Universität München (TUM), 85748 Garching, Germany burschka at cs.tum.edu

plane as $r = \begin{pmatrix} \frac{x_P}{z_P} & \frac{y_P}{z_P} & 1 \end{pmatrix}^T$ and the unit length vector as $\bar{r}$. In the following perturbation analysis we assume that the tracking errors of different features and different components are uncorrelated and have the same standard deviation σ. The Essential matrix, E, describes the mapping between two representations of a point in different camera coordinate frames, (P, P'), such that $r'^T E r = 0$ and $E = [t]_\times R$, where $P' = RP + t$ with R denoting the rotation matrix and t representing the translation vector. For the perturbation analysis we assume an additive translational error $\tilde{t} = t + \delta_t$. However, the rotation error has to meet some requirements in order to fulfil the orthogonality constraint of a rotation matrix [2]:

$$\begin{aligned} I &= (R + \Delta_R)(R + \Delta_R)^T \\ &= RR^T + \Delta_R R + R\Delta_R^T + \mathscr{O}\left(\Delta_R \Delta_R^T\right) \\ &\cong I + \Delta_R R + R\Delta_R^T \\ &\Rightarrow \Delta_R R^T = -\left(\Delta_R R^T\right)^T \overset{def}{=} \Delta'_R = \left[d'_R\right]_\times \end{aligned} \tag{1}$$

where the operator $[\cdot]_\times$ denotes the cross-product skew matrix. It follows that

$$R + \Delta_R = \left(I + \Delta_R R^T\right) R = \left(I + \Delta'_R\right) R \overset{def}{=} R_{\Delta'_R} R \,. \tag{2}$$

The covariance matrix of a noise vector can then be computed by the expectation of its tensor product, $\Gamma_x = \mathbb{E}\left(\delta_x \delta_x^T\right)$, considering only the first order errors with zero mean.

The following theorem will be used in several steps of the derivations.

Theorem: Let $A = H \operatorname{diag}(\lambda_1, \ldots, \lambda_n) H^T$ be an $n \times n$ symmetric matrix where diag() denotes a diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_n$ as entries and H is an orthonormal matrix consisting of the eigenvectors $h_1, \ldots, h_n$. The perturbation of a scaled eigenvector $x = kh_i$, with scaling factor $k \in \mathbb{R}$, can then be described up to first order by the perturbation in A, denoted as Δ_A, according to $\delta_x \cong H D_i H^T \Delta_A x$ with $D_i = \operatorname{diag}(l_1, \ldots, l_n)$ and $l_j = \begin{cases} 0 & \text{if } i = j \\ \frac{1}{\lambda_i - \lambda_j} & \text{else} \end{cases}$. *Proof*: See [14].

III. Z∞ ALGORITHM

The Z∞ algorithm uses the generally undesirable quantization effect of digital cameras to separate translation-invariant from translation-dependent landmarks [8]. Landmarks, which are far enough that the actual translation is not measurable, are projected on the same pixel location before and after translation. Such feature correspondences represent only the rotational component of the motion. This enables the Z∞ algorithm to determine the rotational component of the motion without ever considering the translational one. Hence, the solution space has less dimensions, which simplifies the computation and the suppression of outliers significantly. The key problem, namely to identify translation-invariant landmarks for rotation estimation, is solved in a RANSAC framework [3]. Of course, the approach is only applicable if enough features have a camera and translation dependent minimum distance, which is in general the case for outdoor or slow indoor applications with wide angle cameras.

A. Rotation Estimation

The rotation estimation of the Z∞ algorithm is based on the direction vectors to different landmarks and uses the Umeyama algorithm described in [13] to compute the DCM based on a SVD. The corresponding unit-length image rays used for rotation estimation belong all to the set of translation-invariant points, $\mathscr{R}_{\text{inl}} = \left\{(r_i, r'_i) \,|\, \bar{r}'_i = R^T \bar{r}_i\right\}$. Thus, following least squares problem for an initially unknown set $\mathscr{S}$ has to be solved

$$\mathrm{R}(\mathscr{S}) : \quad R = \arg\min_R \left(\sum_{(\bar{r}_i, \bar{r}'_i) \in \mathscr{S}} \left\| R^T \bar{r}_i - \bar{r}'_i \right\| \right) . \tag{3}$$

For that, the origin of the coordinate frame has to be moved to the center of the point cloud, resulting in

$$r_i^* = \bar{r}_i - c \quad | \quad c = \frac{1}{|\mathscr{S}|} \sum_{\bar{r}_i \in \mathscr{S}} \bar{r}_i \tag{4}$$

and $r_i'^*$ respectively. The non-scaled sample cross-covariance matrix $M = \sum r_i'^* r_i^{*T}$ for these point clouds is calculated from the set $\mathscr{S}^* = \left\{(r_i^*, r_i'^*)\right\}$, such that

$$R = V_M W U_M^T \quad \text{with} \quad W = \operatorname{diag}\left(1, 1, \det\left(V_M U_M^T\right)\right) \tag{5}$$

and $\operatorname{SVD}(M) = U_M \Sigma_M V_M^T$, while the function det() computes the determinant.

B. Translation Estimation

All the remaining feature correspondences in the outlier set $\mathscr{R}_{\text{outl}}$ are rotated back by R and are then only affected by the translational part of the motion

$$\mathring{r}' = R r' = RR^T (r - t) = r - t \tag{6}$$

with $t \overset{!}{\neq} 0$ because only translation-dependent features are used. Projecting the rays on the unit focal image plane we get $\check{r}' = \frac{\mathring{r}'}{z_{\mathring{r}'}}$. The back-rotated optical flow vectors $\left\{(r_i \check{r}'_i)\right\}$ should all meet in the epipole q, which means that the epipole is the intersection of all the lines defined by the translational flow vectors. This constraint defines the following linear system of equations

$$A q = d \quad \Rightarrow \quad \begin{pmatrix} A & d \end{pmatrix} \begin{pmatrix} q \\ -1 \end{pmatrix} = 0 \overset{def}{=} B g \tag{7}$$

with $d = \begin{pmatrix} n_1^T r_{1;1:2} & \ldots & n_{|\mathscr{R}_{\text{out}}|}^T r_{|\mathscr{R}_{\text{out}}|;1:2} \end{pmatrix}^T$, $A = \begin{pmatrix} n_1 & \ldots & n_{|\mathscr{R}_{\text{out}}|} \end{pmatrix}^T$ and $n_i = \begin{pmatrix} y_{\check{r}'_i} - y_{r_i} & x_{r_i} - x_{\check{r}'_i} \end{pmatrix}^T$.

Thus, the direction of translation up to sign $\tilde{t}$ can be computed by

$$\tilde{t} = \frac{\tilde{t}'}{\|\tilde{t}'\|} \quad \text{with} \quad \tilde{t}' = \begin{pmatrix} q \\ 1 \end{pmatrix}, q = -\frac{V_{B;1:2,3}}{V_{B;3,3}} \tag{8}$$

and $\operatorname{SVD}(B) = U_B \Sigma_B V_B^T$. The ambiguity of sign of the translation vector can be resolved by

$$t = \begin{cases} -\tilde{t} & \text{if } \sum_i (r_i \times \check{r}'_i)^T (\tilde{t} \times r_i) < 0 \\ \tilde{t} & \text{else} \end{cases} . \tag{9}$$

C. Error Propagation of the Rotation Estimation

The first order accuracy analysis of the orthogonal Procrustes problem, which determines the rotation between two corresponding point clouds, has been presented and discussed in [2]. We adapt this approach to provide a first order error propagation for the Z_∞ rotation estimation.

Using the rotation error definition as introduced in Eq. 2, the column vector δ_R of the matrix Δ_R can be computed from $[d'_R]_\times$ by

$$\delta_R = \operatorname{col}\left[[-R_{-,i}]_\times\right]_{i=1}^{3} d'_R \overset{def}{=} G_R\, d'_R \tag{10}$$

with $\operatorname{col}[\cdot]$ representing the vertical concatenation. As derived in [2] the covariance matrix for d'_R can be estimated by

$$\begin{aligned}\Gamma_{d'_R} = -L\Bigg(&\sum_{i=1}^{N}\left([Rr_i^*]_\times \Gamma_{r_i'^*} [Rr_i^*]_\times\right) + \\ &\sum_{i=1}^{N}\left(\left[r_i'^*\right]_\times R\Gamma_{r_i^*}R^T\left[r_i'^*\right]_\times\right)\Bigg)L\end{aligned} \tag{11}$$

with $L = RHR^T$ and $H = (\operatorname{tr}(S)I - S)^{-1}$ where $S = V_M\Sigma_M V_M^T$ according to Eq. 5. The vectors r_i^* and $r_i'^*$ were introduced in Eq. 4 and $\Gamma_{r_i^*}$ and $\Gamma_{r_i'^*}$ are the corresponding covariance matrices. These can be computed by weighting the feature detector variance σ^2 by the feature ray lengths, such that $\Gamma_{r_i^*} = \left(\frac{z_{r_i}}{f}\right)^2\Gamma_{\tilde{r}}$ and $\Gamma_{r_i'^*} = \left(\frac{z_{r_i'}}{f}\right)^2\Gamma_{\tilde{r}'}$ with f being the focal length in pixels and $\Gamma_{\tilde{r}} = \Gamma_{\tilde{r}'} = \operatorname{diag}(\sigma^2, \sigma^2, 0)$. The shift of the origin to the centroid does not affect the variance. Finally, putting together Eq. 10 and 11 we get $\Gamma_R = G_R\Gamma_{d'_R}G_R^T$.

D. Error Propagation of the Translation Estimation

The error in the translation estimation has to be propagated through Eq. 7. The initial error Δ_B of B can be estimated according to $\Delta_B = \begin{pmatrix}\Delta_A & \delta_d\end{pmatrix}$ with $\Delta_A = \begin{pmatrix}\delta_{n_1} & \dots & \delta_{n_{|\mathscr{R}_\text{out}|}}\end{pmatrix}^T$ and $\delta_d \cong \begin{pmatrix}\delta_{d_1} & \dots & \delta_{d_{|\mathscr{R}_\text{out}|}}\end{pmatrix}^T$ where

$$\begin{aligned} n + \delta_n &= \begin{pmatrix}\left(y_{\check{r}_i'} + \delta_{y_{\check{r}_i'}}\right) - \left(y_{r_i} + \delta_{y_{r_i}}\right) \\ \left(x_{r_i} + \delta_{x_{r_i}}\right) - \left(x_{\check{r}_i'} + \delta_{x_{\check{r}_i'}}\right)\end{pmatrix} \Rightarrow \\ \delta_n &= \begin{pmatrix}\delta_{y_{\check{r}_i'}} - \delta_{y_{r_i}} \\ \delta_{x_{r_i}} - \delta_{x_{\check{r}_i'}}\end{pmatrix} \qquad \text{and}\end{aligned} \tag{12}$$

$$\begin{aligned} d + \delta_d &= (n + \delta_n)^T\begin{pmatrix}x_r + \delta_{x_r} \\ y_r + \delta_{y_r}\end{pmatrix} \quad \Rightarrow \\ \delta_d &\cong x_n\delta_{x_r} + x_{\delta_n}x_r + y_n\delta_{y_r} + y_{\delta_n}y_r\,.\end{aligned} \tag{13}$$

The error propagation from the feature-error to δ_{B^T} can be computed by following transition matrices:

$$\begin{aligned}\delta_{r,n} &= \begin{pmatrix}\delta_{r_1} & \delta_{n_1} & \dots & \delta_{r_{|\mathscr{R}_\text{out}|}} & \delta_{n_{|\mathscr{R}_\text{out}|}}\end{pmatrix}^T \\ &= \left(I_{|\mathscr{R}_\text{out}|} \otimes \begin{pmatrix}1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0\end{pmatrix}\right)\begin{pmatrix}\delta_{r_1} \\ \delta_{\check{r}_1'} \\ \vdots \\ \delta_{r_{|\mathscr{R}_\text{out}|}} \\ \delta_{\check{r}'_{|\mathscr{R}_\text{out}|}}\end{pmatrix} \\ &\overset{def}{=} G_{r,n}\,\delta_{r,\check{r}'} \quad \text{and}\end{aligned} \tag{14}$$

$$\begin{aligned}\delta_{B^T} &\cong \operatorname{Diag}\left(P_1, P_2, \dots, P_{|\mathscr{R}_\text{out}|}\right)\delta_{r,n} \overset{def}{=} G_{B^T}\,\delta_{r,n} \\ &\text{with} \quad P_i = \begin{pmatrix}0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ x_{n_i} & y_{n_i} & x_{\tilde{r}_i} & y_{\tilde{r}_i}\end{pmatrix}\end{aligned} \tag{15}$$

and Diag() denoting a block-diagonal matrix. The error of $\tilde{t}$, which is derived from the least-squares solution of Eq. 7 according to Eq. 8, can be propagated using the theorem introduced in Section II. Thus, we first need to compute the transition matrix from the error vector δ_{B^T} to δ_{B^TB}

$$\begin{aligned}\delta_{B^TB} &\cong \left(\operatorname{row}\left[I_3 \otimes B_{i,-}^T\right]_{i=1}^{|\mathscr{R}_\text{out}|} + B^T \otimes I_3\right)\delta_B \\ &\overset{def}{=} G_{B^TB}\,\delta_{B^T}\end{aligned} \tag{16}$$

with $\operatorname{row}[\cdot]$ representing the horizontal concatenation. Now, the perturbation of the eigenvector can be propagated by

$$\begin{aligned}\delta_{\tilde{q}} &\cong V_B D_{B^TB;3} V_B^T \Delta_{B^TB}\begin{pmatrix}q \\ -1\end{pmatrix} \\ &= V_B D_{B^TB;3} V_B^T\left(\begin{pmatrix}q \\ -1\end{pmatrix}^T \otimes I_3\right)\delta_{B^TB} \\ &\overset{def}{=} G_{\tilde{q}}\,\delta_{B^TB}\,,\end{aligned} \tag{17}$$

where q has been introduced in Eq. 8. According to those equations, the error of $\delta_{\tilde{t}}$ of the translation axis $\tilde{t}$ can be estimated from $\delta_{\tilde{q}}$ by $\delta_{\tilde{t}} = \begin{pmatrix}\frac{\delta_{\tilde{q}}}{\|\tilde{r}'\|} \\ 0\end{pmatrix}$ and, thus, $\delta_{\tilde{t}} = G_t\,\delta_{\tilde{q}}$ with $G_t = \operatorname{diag}(z_{\tilde{t}}, z_{\tilde{t}}, 0)$. The error for t and $\tilde{t}$ varies only in the sign and, hence, results in the same covariance propagation matrix.

It remains to compute the initial covariance matrix $\Gamma_{r,\check{r}'} = \mathbb{E}\left(\delta_{r,\check{r}'}\delta_{r,\check{r}'}^T\right)$, with $\delta_{r_i,\check{r}_i'} = \frac{\sigma}{f}I_4$, which yields $\Gamma_{r,\check{r}'} = \left(\frac{\sigma}{f}\right)^2 I_{(4\cdot|\mathscr{R}_\text{out}|)}$.

Putting together the pieces we can compute the covariance matrix of the direction of translation vector, Γ_t, such that

$$\Gamma_t \cong H_t\,\Gamma_{r,\check{r}'}\,H_t^T \tag{18}$$

with $H_t = G_t\,G_{\tilde{q}}\,G_{B^TB}\,G_{B^T}\,G_{r,n}$.

IV. Eightpoint Algorithm

The 8-point algorithm is a closed-form solution which computes the Essential matrix E by solving a linear system of equations [7], defined as $Ae = 0$ where e is the vectorization of E, $e = \operatorname{vec}(E)$, and $A \in \mathbb{R}^{N\times 9}$ is a stack of N row vectors, where N denotes the number of features.

A. Essential Matrix Estimation

The least-squares solution for $e = \arg\min_e (\|Ae\|)$ is found by the last column of V_A corresponding to the smallest singular value, such that

$$e = V_{A;-,9} \quad \text{where} \quad \mathrm{SVD}(A) = U_A \Sigma_A V_A^T \,, \tag{19}$$

and the indices are denoted by the succeeding subscript. A major improvement of the conditioning of the eight-point algorithm was first proposed by Hartley [4]. Introducing two transformation matrices, S and S', we are able to manipulate the input of the algorithm, which improves the computational condition for estimating E:

$$r'^T E r = \left(S' r'\right)^T S'^{-T} E S^{-1} \, S r = \hat{r}'^T \hat{E} \hat{r} = 0 \,. \tag{20}$$

The matrix E can then be retrieved by $E = S'^T \hat{E} S$. In order to make the computation as robust as possible according to total least squares estimation, the following matrices for S and S' are proposed by Mühlich [9]. S provides an anisotropic normalization of r and is defined as

$$S = \mathrm{chol}(M)^{-1} \quad \text{with} \quad M = \frac{1}{N} \sum_{i=1}^{N} r_i r_i^T \,, \tag{21}$$

where the Cholesky decomposition is denoted by $\mathrm{chol}(M) = K$ with K being an upper triangular matrix such that $M = KK^T$. S' describes a shift of the origin and an isotropic scaling for r', such that

$$\begin{aligned} S' &= \begin{pmatrix} s & 0 & -sc_1 \\ 0 & s & -sc_2 \\ 0 & 0 & 1 \end{pmatrix} \text{ with} \\ c = \frac{1}{N} \textstyle\sum_{i=1}^{N} r'_{i;1:2} \quad &\text{and} \quad s = \frac{\sqrt{2}}{\frac{1}{N} \sum_{i=1}^{N} \|r'_{i;1:2} - c\|} \,. \end{aligned} \tag{22}$$

Thus, c denotes the centroid and s an isotropic scaling factor averaging the length of each point to the length of $\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$.

Next, we enforce the constraint of the Essential Matrix, such that

$$\bar{E} = U_E \Sigma_{\bar{E}} V_E^T = U_E \, \mathrm{diag}(1,1,0) \, V_E^T \tag{23}$$

with $\mathrm{SVD}(E) = U_E \Sigma_E V_E^T$.

B. Decomposition of the Essential Matrix

Once the Essential matrix has been computed, it can be decomposed into translation and rotation. By modifying the polar decomposition of $\bar{E}$ we yield

$$\begin{aligned} \bar{E} &= U_E \Sigma_{\bar{E}} V_E^T \\ &= U_E \Sigma_{\bar{E}} \, WZU_E^T \quad U_E ZW \, V_E^T \\ &= [t]_\times R \quad \text{with} \\ W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det\left(U_E V_E^T\right) \end{pmatrix} &, \quad \tilde{Z} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \text{and} \quad Z &= \begin{cases} \tilde{Z}^T & \text{if } \left(\sum_i A_i^+ t\right)^T 1_2 < 0 \\ \tilde{Z} & \text{else} \end{cases} \,, \end{aligned} \tag{24}$$

where $A_i = \begin{pmatrix} -Rr_i & r'_i \end{pmatrix}$ and A_i^+ denotes the pseudoinverse of A_i.

The translation vector can be found as the unit vector minimizing $\tilde{t} = \arg\min_t \left(\|\bar{E}^T t\|\right)$ which corresponds to finding the unit eigenvector of $\bar{E}\bar{E}^T$ associated with the smallest eigenvalue $\tilde{t} = U_{E;-,3}$.

It remains to solve for the ambiguity of sign by

$$t = \begin{cases} -\tilde{t} & \text{if } \sum_i \left(Rr_i \times r'_i\right)^T \bar{E} r_i < 0 \\ \tilde{t} & \text{else} \end{cases} \,. \tag{25}$$

C. Error Propagation of the Essential Matrix Estimation

The first order perturbation analysis of the 8-point algorithm is adapted from [14]. However, in our derivation we take also into consideration the modifications proposed by Hartley and Mühlich and the computation of the DCM at the decomposition of the Essential matrix as well as the error axis representation, which allows for geometric insights.

Let us assume a measurement matrix $\tilde{A}$, which is the sum of the matrix A denoting the true value and the perturbation matrix Δ_A. According to [14], the first-order error of the non-normalized Essential matrix $\hat{E}$ can be computed from the transposed of the perturbation matrix $\delta_{A^T} = \mathrm{vec}\left(\Delta_A^T\right)$ by

$$\delta_{\hat{e}} \cong V_A D_{A^T A;9} V_A^T \left(\hat{e}^T \otimes I_9\right) \delta_{A^T A} \overset{def}{=} G_{\hat{E}} \delta_{A^T A} \tag{26}$$

with $\mathrm{SVD}(A) = U_A \Sigma_A V_A^T$ and $\hat{e}$ being the vectorization of $\hat{E}$, whereas

$$\begin{aligned} \delta_{A^T A} &\cong \left(A^T \otimes I_9 + \mathrm{row}\left[I_9 \otimes (A_{i,-})^T\right]_{i=1}^{N}\right) \delta_{A^T} \\ &\overset{def}{=} G_{A^T A} \, \delta_{A^T} \,. \end{aligned} \tag{27}$$

The result for the non-normalized features can then be recovered by $E = S'^T \left(\hat{E} + \Delta_{\hat{E}}\right) S$ which leads to $\Delta_E = S'^T \Delta_{\hat{E}} S$ and, finally, to

$$\begin{aligned} \delta_e &= \left(S^T \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}\right) \circ \left(\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \otimes S'^T\right) \delta_{\hat{e}} \\ &\overset{def}{=} G_E \, \delta_{\hat{e}} \,. \end{aligned} \tag{28}$$

Knowing the propagation of the error from the input matrix A to the Essential matrix, we can estimate the covariance matrix of e by

$$\Gamma_E = H_E \Gamma_{A^T} H_E^T \quad \text{with} \quad H_E = G_E G_{\hat{E}} G_{A^T A} \,. \tag{29}$$

To consider the normalization effect of the 8-point derivatives we need to adapt the measurement error for each feature, such that $\Gamma_{\hat{r}} = \left(\mathrm{diag}\left(\frac{1}{f} S \delta_{\tilde{r}}\right)\right)^2$ and $\Gamma_{\hat{r}'} = \left(\mathrm{diag}\left(\frac{1}{f} S' \delta_{\tilde{r}'}\right)\right)^2$. This results in $\Gamma_{\hat{A}^T} = \mathrm{Diag}\left(\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_N\right)$, with f being the focal length used to scale the error and $\hat{P}_i = \left(\hat{r}\hat{r}^T\right) \otimes \Gamma_{\hat{r}'} + \Gamma_{\hat{r}} \otimes \left(\hat{r}'\hat{r}'^T\right)$. In Eq. 23 the Frobenius norm of the Essential matrix is set to 2. This step is approximated by adjusting the Frobenius norm of the covariance matrix to the Frobenius norm of $\bar{E}$ which yields

$$\Gamma_{\bar{E}} \approx H_E \Gamma_{\hat{A}^T} H_E^T \frac{2}{\sum_{i=1}^{3} \sigma_{\hat{A};i}^2} \tag{30}$$

with $\mathrm{SVD}\left(\hat{A}\right) = U_{\hat{A}} \, \mathrm{diag}\left(\sigma_{\hat{A};1}, \sigma_{\hat{A};2}, \sigma_{\hat{A};3}\right) V_{\hat{A}}^T$ and $\hat{A}$ denoting the matrix resulting from the normalized features, $\hat{r}$ and

$\hat{r}'$. Please note that this is only an approximation and does actually not correspond to Eq. 23, because it is not assured that the singular values of $\bar{E}$ are $\mathrm{diag}(1,1,0)$ but only that the Frobenius norm is 2.

D. Error Propagation of the Essential Matrix Decomposition

According to Eq. 24, the rotation matrix is estimated by $R = U_E Z W V_E^T$. Consequently the error of R propagates as follows

$$\begin{aligned} R + \Delta_R &= (U_E + \Delta_{U_E}) ZW (V_E + \Delta_{V_E})^T \Rightarrow \\ \Delta_R &\cong \Delta_{U_E} ZWV_E^T + U_E ZW \Delta_{V_E}^T \ . \end{aligned}$$

Hence, we need to propagate the error for Δ_{U_E} and Δ_{V_E}. The vectors of U_E and V_E are eigenvectors of $\bar{E}\bar{E}^T$ and $\bar{E}^T\bar{E}$ respectively. Thus, we first need to compute the transition matrix $G_{\bar{E}\bar{E}^T}$ and $G_{\bar{E}^T\bar{E}}$ by

$$\begin{aligned} \delta_{\bar{E}\bar{E}^T} &\cong \left(\mathrm{row}\,[I_3 \otimes \bar{E}_{-,i}]_{i=1}^3 + \bar{E} \otimes I_3\right) \delta_e \\ &\overset{def}{=} G_{\bar{E}\bar{E}^T}\, \delta_e \quad \text{and} \end{aligned} \tag{31}$$

$$\begin{aligned} \delta_{\bar{E}^T\bar{E}} &\cong \left(\mathrm{col}\left[I_3 \otimes (\bar{E}_{-,i})^T\right]_{i=1}^3 + I_3 \otimes \bar{E}^T\right) \delta_e \\ &\overset{def}{=} G_{\bar{E}^T\bar{E}}\, \delta_e \ . \end{aligned} \tag{32}$$

Now, the error in U_E and V_E can be computed using again the theorem introduced in Section II

$$\begin{aligned} \delta_{U_{\bar{E}}} &\cong \mathrm{col}\left[U_E D_{\bar{E}\bar{E}^T;i} U_E^T \left((U_{E;-,i})^T \otimes I_3\right)\right]_{i=1}^3 \delta_{\bar{E}\bar{E}^T} \\ &\overset{def}{=} G_{U_{\bar{E}}}\, \delta_{\bar{E}\bar{E}^T} \quad \text{and} \end{aligned} \tag{33}$$

$$\begin{aligned} \delta_{V_{\bar{E}}} &\cong \mathrm{col}\left[V_E D_{\bar{E}^T\bar{E};i} V_E^T \left((V_{E;-,i})^T \otimes I_3\right)\right]_{i=1}^3 \delta_{\bar{E}^T\bar{E}} \\ &\overset{def}{=} G_{V_{\bar{E}}}\, \delta_{\bar{E}^T\bar{E}} \ , \quad \text{where} \end{aligned} \tag{34}$$

$D_{\bar{E}\bar{E}^T;1} = D_{\bar{E}\bar{E}^T;2} = D_{\bar{E}^T\bar{E};1} = D_{\bar{E}^T\bar{E};2} = \mathrm{diag}(0,0,1)$ as proposed by [5], to avoid a division by zero.

Finally, the perturbation in the rotation matrix can be computed by

$$\begin{aligned} \delta_R &= \begin{pmatrix} G_{RU} & G_{RV} \end{pmatrix} \begin{pmatrix} \delta_{U_{\bar{E}}} \\ \delta_{V_{\bar{E}}} \end{pmatrix} \overset{def}{=} G_R \begin{pmatrix} \delta_{U_{\bar{E}}} \\ \delta_{V_{\bar{E}}} \end{pmatrix} \text{ with} \\ G_{RU} &= \mathrm{col}\big[\big(Z_{1,2} V_{E;i,2} \quad Z_{2,1} V_{E;i,1} \\ &\qquad \det\left(U_E V_E^T\right) V_{E;i,3} \otimes I_3\big)\big]_{i=1}^3 \quad \text{and} \\ G_{RV} &= \big(I_3 \otimes Z_{2,1} U_{E;i,2} \quad I_3 \otimes Z_{1,2} U_{E;i,1} \\ &\qquad I_3 \otimes \det\left(U_E V_E^T\right) U_{E;i,3}\big) \ . \end{aligned} \tag{35}$$

This yields the following covariance for the rotation matrix

$$\begin{aligned} \Gamma_R &= \begin{pmatrix} G_{RU}\, G_{U_{\bar{E}}} & G_{RV}\, G_{V_{\bar{E}}} \end{pmatrix} \begin{pmatrix} G_{\bar{E}\bar{E}^T} \\ G_{\bar{E}^T\bar{E}} \end{pmatrix} \Gamma_{\bar{E}} \\ &\quad \begin{pmatrix} G_{\bar{E}\bar{E}^T}^T & G_{\bar{E}^T\bar{E}}^T \end{pmatrix} \begin{pmatrix} G_{U_{\bar{E}}}^T\, G_{RU}^T \\ G_{V_{\bar{E}}}^T\, G_{RV}^T \end{pmatrix} . \end{aligned} \tag{36}$$

Geometric insights of the rotation error can be gained computing the error rotation axis d'_R. It can be derived as the eigenvector of Δ'_R corresponding to the smallest eigenvalue of $Q = \Delta'_R \Delta'^T_R = \Delta_R \Delta_R^T$ weighted by the mean of the square roots of the two largest singular values, $\sigma_m = 0.5\left(\sqrt{\sigma_{Q;1}} + \sqrt{\sigma_{Q;2}}\right)$, resulting in $d'_R = \sigma_m U_{Q;-,3}$, whereas $\mathrm{SVD}(Q) = U_Q \,\mathrm{diag}(\sigma_{Q;1}, \sigma_{Q;2}, \sigma_{Q;3})\, V_Q^T$ and $Q = \Gamma_{R;1:3,1:3} + \Gamma_{R;4:6,4:6} + \Gamma_{R;7:9,7:9}$.

The translation vector equals up to sign to the eigenvector corresponding to the smallest eigenvalue of $\bar{E}\bar{E}^T$. Analogue to Eq. 26, the first order perturbation propagation of $\tilde{t}$ can then be computed using Eq. 31 by

$$\delta_{\tilde{t}} \cong U_E D_{\bar{E}\bar{E}^T;3} U_E^T \left(\tilde{t}^T \otimes I_3\right) \delta_{\bar{E}\bar{E}^T} \overset{def}{=} G_{\tilde{t}}\, \delta_{\bar{E}\bar{E}^T} \tag{37}$$

with $\mathrm{SVD}(\bar{E}) = U_E \Sigma_{\bar{E}} V_E^T$. This yields the following covariance matrix, which is equal to the one for t because the transition matrix $G_{\tilde{t}}$ is applied twice:

$$\Gamma_t = \Gamma_{\tilde{t}} = G_{\tilde{t}}\, G_{\bar{E}\bar{E}^T} \Gamma_{\bar{E}}\, G_{\bar{E}\bar{E}^T}^T\, G_{\tilde{t}}^T \ . \tag{38}$$

V. Experiments

In the following experiments we evaluate the accuracy of the presented first-order error propagation and compare the performance of the Z_∞ and the 8-point algorithm. We implemented three 8-point variants, which we named according to their inventors. The *original 8-point (Longuet-Higgins variant)* consists of the unmodified image rays and, thus, uses the non-normalized A-matrix. The *Hartley variant* uses isotropic scaling for both sets of image rays, S and S', as denoted in Eq. 22. The *Mühlich variant* provides anisotropic scaling for the first image rays and isotropic scaling for the rays of the second image as introduced in Eq. 21 and 22. We are not showing any results for the fixed-column perturbation estimation also introduced by Mühlich, which only leads to notable performance improvements in the case of error free features in the reference image.

Unfortunately, in practice we can only use the error-corrupted $\tilde{A}$ for the error propagation analysis, because it is the value we measure. However, from a slightly different point of view, we can regard A as noise-corrupted matrix by adding $-\Delta_A$ to the matrix $\tilde{A}$. Thus, the error is the deviation of the true solution from the noise-corrupted one. This observation justifies the use of $\tilde{A}$ to estimate the errors [14].

To compare the real and the estimated error as a scalar we compute the errors of the estimates f_E, f_R, f_t and the propagated errors $\hat{f}_E$, $\hat{f}_R$, $\hat{f}_t$ by normalizing the Frobenius norm of the error by the Frobenius norm of the respective matrix

$$f_E = \frac{1}{\sqrt{2}} \|E_S - \bar{E}\|_F \ , f_R = \frac{1}{\sqrt{3}} \|R_S - R\|_F \ , f_t = \|t_S - t\|_F \tag{39}$$

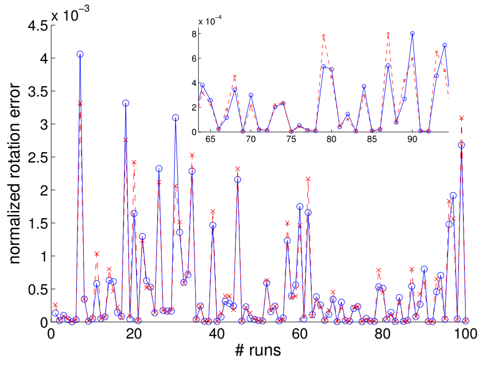
with the subscript $_S$ denoting the simulated values. Furthermore, an equivalent to the normalized Frobenius norm can be computed from the covariance matrices by

$$\hat{f}_E = \sqrt{\frac{\mathrm{tr}(\Gamma_E)}{2}} \ , \ \hat{f}_R = \sqrt{\frac{\mathrm{tr}(\Gamma_R)}{3}} \text{ and } \hat{f}_t = \sqrt{\mathrm{tr}(\Gamma_t)} \ . \tag{40}$$

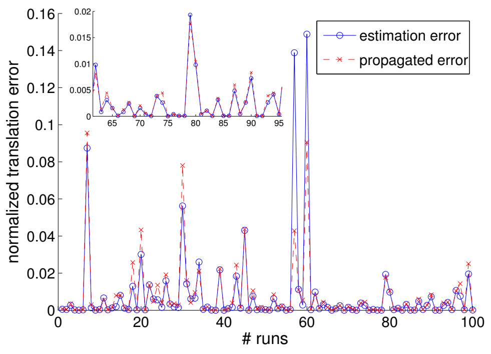
Similar results were achieved by comparing the simulated and estimated absolute error angles $\|d'_R\|$.

In order to provide an exact ground truth which allows to evaluate the first-order approximation accuracy for the algorithms, we simulated landmarks and camera motions.

The simulated camera had a resolution of 600×600 pixel, the rotation was 5° about a random axis, the translational distance along an arbitrary direction was 5 m and the landmarks were randomly distributed in the field of view within a distance of up to 50 m. We chose 100 random combinations of aperture angle, number of features and noise-level of the features in the image, which is related to the accuracy of the feature detection method. The number of feature correspondences was in the range of 10 to 500, the noise was from 0.01 to 2 pixels and the aperture angle was between 10° and 170°. Assuming tracking-by-matching techniques, like in PTAM [6], we simulated noise for the features in both images. For each of these combinations, we ran 10 cycles with randomly chosen landmark locations, yielding 1000 runs in total.



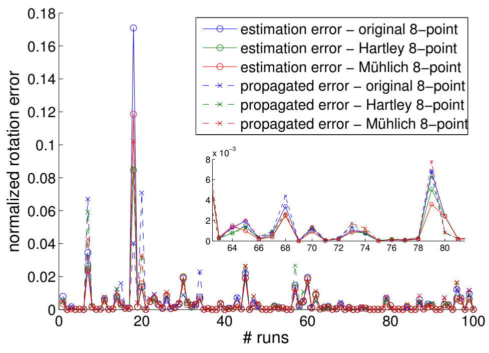
(a) Z_∞ rotation matrix (R)



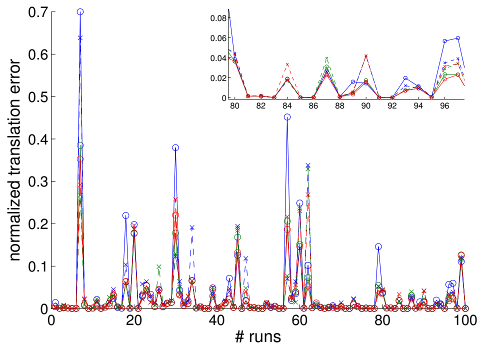
(b) Z_∞ translation vector (t)

Fig. 1. These plots depict the Z_∞ estimation and propagation error for both motion components. Clippings containing small errors are enlarged to show also the correlations at lower error scales.

Fig. 1 and 2 show the median error for each of the 100 parameter combinations and for both algorithms. Both, the error of the motion estimation and the estimated error by the presented uncertainty propagation are shown. The plots prove that the estimated errors correlate, even at smaller scales. However, in some runs, the propagation differs clearly from the effective error, which can be explained by the first-order approximation in the propagation. Furthermore, the uncertainty in the feature locations is only known by the noise variance, but the effective error of each individual landmark projection in the simulation is picked randomly from the corresponding normal distribution.

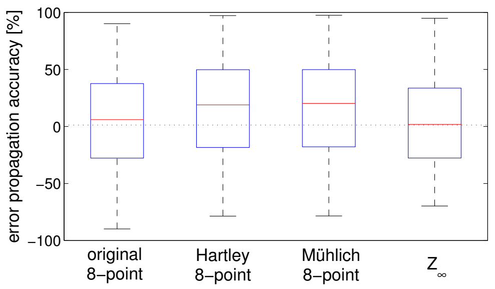


(a) 8-point rotation matrix (R)

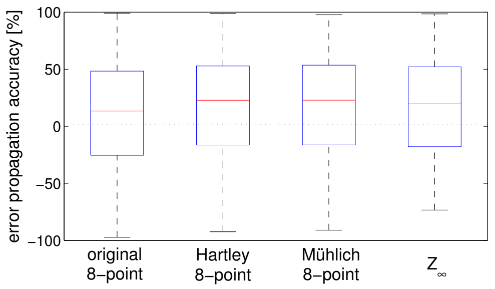


(b) 8-point translation vector (t)

Fig. 2. These plots show the estimation error and the propagated error for the rotation and the translation estimation of the 8-point algorithm variants. Please note the different scales of the y-axes in the Z_∞ and 8-point diagrams.



(a) rotation matrix (R)



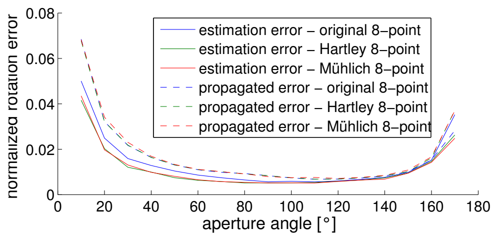
(b) translation vector (t)

Fig. 3. The boxplots in these two figures describe the distribution of the differences between the estimation and the propagation error, computed by Eq. 41 for all 1000 simulation runs.

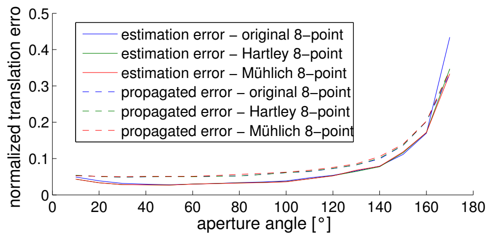
The correlation between the estimation and the propagation error is summarized in the boxplots of Fig. 3. These illustrate the distribution of the differences in all 1000 runs. The differences, d_x, were computed according to the following equation:

$$d_x = \begin{cases} \frac{\hat{f}_x - f_x}{f_x} & \text{if } f_x > \hat{f}_x \\ \frac{\hat{f}_x - f_x}{\hat{f}_x} & \text{else} \end{cases} \quad | x \in \{R, t\} \ . \qquad (41)$$

The Hartley and the Mühlich variation overestimates the error more than the other methods, which can be explained by the additional approximation in Eq. 30. However, an overestimation of the error is less problematic than its underestimation in terms of the fusion with other sensor information. The boxplots also show that the error propagation can fail sometimes. However, it is still favorable to use an uncertainty measure which correlates with the effective error rather than to use the same uncertainty value for all estimates – especially considering that the errors of the visual motion estimation can vary by several orders of magnitude depending on the camera parameters and the landmark locations, as Fig. 1 and 2 reveal.



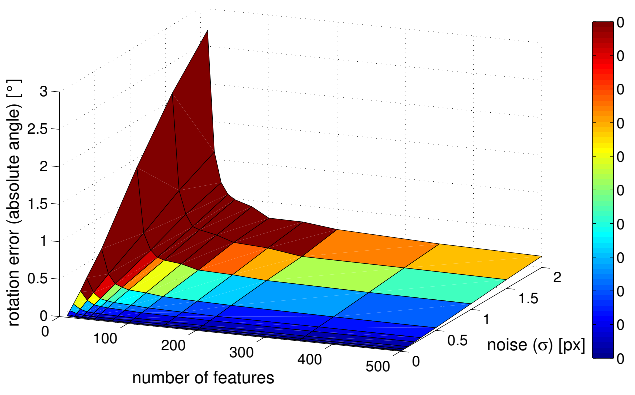
(a) rotation matrix (R)



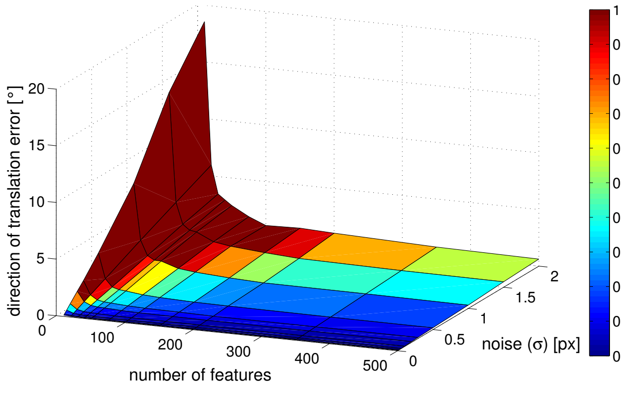
(b) translation vector (t)

Fig. 4. This diagrams show the correlation of the 8-point error propagation for a varying aperture angle, where 10 features and pixel accurate feature tracking were simulated.

We also evaluated the accuracy of the error propagation separately for each of the parameters: aperture angle, number of features and tracking accuracy. The results prove a proper correlation of the error propagation for all parameters. For the sake of space, we show only two plots based on the 8-point algorithm (similar results were achieved with the Z_∞ algorithm), which illustrate some interesting insights. In Fig. 4 the variation of the aperture angle for all 8-point variations is plotted. We took the median value of 100 runs with random landmark locations for each aperture angle, where all the other parameters were considered constant. Interestingly, the plot reveals that the conditioning of the motion estimation does not increase continuously with the aperture angle, but it has a flat optimum around 100°. This can be explained by the fact, that the uncertainty of a feature based pose estimation is composed by the number of features, the tracking accuracy and the location of the landmarks in the image, which affects the conditioning of the computation. In the aforementioned experiment we varied the aperture angle, while keeping the number of pixels constant. Hence, a shorter focal length does not only change the tracking accuracy, due to a lower angular resolution, but allows also for an improved conditioning of the pose computation. For really small aperture angles, the bad conditioning outweighs the high accuracy in the feature locations. If the field of view increases, the feature accuracy gets reduced, but the conditioning improves. The latter effect has more impact and, hence, the error of the estimate decreases. However, as soon as the landmark rays intersect perpendicularly, the conditioning of the pose calculation saturates and at some point only the reduction of the angular resolution has some effect. This results in the aforementioned flat optimum for the pose estimation accuracy. At this point we want to emphasize, that while the accuracy has a sweet spot, the robustness of the image based pose computation, in general, continuously improves by increasing the field of view.



(a) rotation matrix (R)

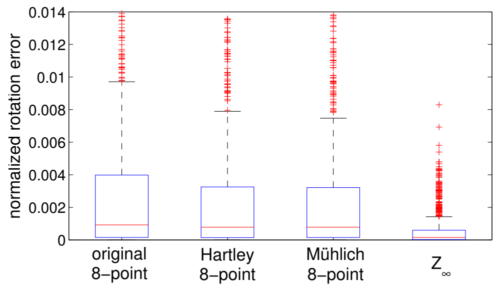


(b) translation vector (t)

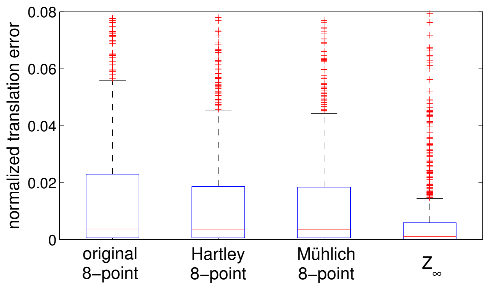
Fig. 5. This plot shows the correlation of the pose estimation accuracy with the tracking accuracy and the number of features for the Mühlich-variant of the 8-point algorithm. We simulated noise in both images and 250 pixels focal length ($\sim$ 100° aperture angle).

The tracking accuracy linearly correlates with the estimation uncertainty, while the number of features has an inverse relationship[1]. Fig. 5 visualizes these correlations. It becomes apparent that for a small number of features (<100) any change of this number yields a significant impact on the performance of the motion computation. Nevertheless, if really high accuracies want to be achieved, it is not sufficient to solely increase the number of features, but a highly accurate tracking has to be used.

If we compare the errors of the different methods, as done by the error boxplots in Fig. 6, we can clearly see that the modified 8-point algorithms outperform the original one, as it is well known in literature. The original 8-point algorithm seems to be more outlier prone, as Fig. 2(a) and 2(b) reveal. We can further see that the separation of the rotation and translation estimation, as done in the Z_∞ algorithm, improves the motion estimation significantly. Hence, if it is possible to split these two motion components, *e.g.* by other sensor information or due to some assumptions on the environment, they should be estimated separately.



(a) rotation matrix (R)



(b) translation vector (t)

Fig. 6. These boxplots show the estimation error of the rotation matrix and the translation vector for the 8-point variants and the Z_∞ algorithm.

VI. Conclusion

We have derived a first-order error propagation for the Z_∞ algorithm and extended the one for the 8-point algorithm to allow for feature normalization as well as the estimation of the rotation matrix and the error rotation vector. An uncertainty measure is crucial if, *e.g.*, the pose estimation needs to be fused with other information, especially because the uncertainty can vary by several orders of magnitude, as the experiments show. We have evaluated the correlation of the error propagation with the effective estimation error on synthetic data, where the results prove a clear correlation. The resulting equations can be used as reference for any application which requires efficient visual pose estimation and its uncertainty evaluation.

We also discussed the influence of aperture angle, tracking accuracy and number of features on the pose estimation accuracy. A larger number of features can quickly improve the pose estimation accuracy up to a certain point, after which it is more efficient to increase the tracking accuracy for further improvements. We have also shown that there is a flat accuracy optimum for the choice of the aperture angle, which is reached as soon as the landmark rays are able to perpendicularly intersect.

Furthermore, the experiments have shown that a separation of the rotation and the translation estimation, as it is done by the Z_∞ algorithm, is strongly recommended if applicable. In case of the 8-point algorithm the rotation and the translation are estimated simultaneously. A lateral translation can be easily misinterpreted as rotation using a camera with a narrow field of view. An open question in literature is still how the rotation and the translation estimation correlates and how these correlations can be taken into account for the uncertainty propagation. Such studies could also give useful insights how to prevent aforementioned misinterpretations. Another interesting work would be the impact analysis of higher order error terms on the uncertainty propagation.

[1]The standard deviation of an estimate, σ_e, which is based on N same uncorrelated measurements with standard deviation σ, can be computed by $\sigma_e = \frac{\sigma}{\sqrt{N}}$.

References

[1] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *PAMI*, pages 1052–1067, 2007.

[2] L. Dorst. First order error propagation of the procrustes method for 3d attitude estimation. *PAMI*, pages 221–229, 2005.

[3] M. Fischler and R. Bolles. 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.

[4] R. Hartley. In defense of the eight-point algorithm. *PAMI*, 19(6):580–593, 1997.

[5] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[6] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*, pages 1–10. IEEE, 2007.

[7] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in computer vision: issues, problems, principles, and paradigms*, page 61, 1987.

[8] E. Mair and D. Burschka. *Mobile Robots Navigation*, chapter Z_∞ - Monocular Localization Algorithm with Uncertainty Analysis for Outdoor Applications, pages 107 – 130. In-Tech, March 2010.

[9] M. Mühlich and R. Mester. The role of total least squares in motion analysis. *ECCV*, page 305, 1998.

[10] R. Newcombe and A. Davison. Live dense reconstruction with a single moving camera. In *CVPR*, pages 1498–1505. IEEE, 2010.

[11] T. Papadopoulo and M. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. *Computer Vision-ECCV 2000*, pages 554–570, 2000.

[12] F. Sur, N. Noury, M. Berger, et al. Computing the uncertainty of the 8 point algorithm for fundamental matrix estimation. In *19th British Machine Vision Conference-BMVC 2008*, 2008.

[13] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *PAMI*, pages 376–380, 1991.

[14] J. Weng, T. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *PAMI*, 11(5):451–476, 1989.

[15] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27(2):161–195, 1998.