The Probabilistic Normal Epipolar Constraint for Frame-To-Frame Rotation Optimization under Uncertain Feature Positions

Supplementary Material

Dominik Muhle*†, Lukas Koestler*†, Nikolaus Demmel†, Florian Bernard‡ and Daniel Cremers†
†Technical University of Munich ‡University of Bonn

A. Overview

In this supplementary material, we present additional insight into the probabilistic normal epipolar constraint (PNEC) energy function, give more details about the practical implementations of the PNEC rotation estimation, and show further experimental results. We address limitations of the PNEC in Sec. B. Sec. C shows how we use the KLT tracking to extract position covariance matrices. The unscented transform and its use for uncertainty propagation are presented in Sec. D. Sec. E gives geometric insights into the PNEC. We derive the directional limit for the singularities in Sec. F. The self-consistent-field (SCF) optimization is explained in Sec. G. Sec. H gives an overview of the hyperparameters we use in our experiments. The simulated experiments for additional noise types not presented in the main paper are shown in Sec. I. Sec. J presents further results of the simulated experiments with regard to the energy function. Experiments on the influence of anisotropy and wrongly estimated covariance matrices are given in Sec. K and Sec. L. We show further analysis of the results on the KITTI dataset together with an ablation study in Sec. M.

B. Limitations

This section will give further details and explanations, which were not mentioned in the main paper due to the constrained space, regarding the limitations of the proposed method.

As written in the main paper, the PNEC optimization scheme consists of two stages that result in a more involved optimization than for the original normal epipolar constraint (NEC). Due to the iterative optimization and the joint refinement, the PNEC will not be as fast as the NEC, which is shown by the runtime experiment in the main paper. However, the integration of both methods into a RANSAC scheme mitigates the difference between them. Overall, our PNEC and the NEC both run in real-time on the KITTI dataset.

A further limitation of the PNEC is its dependence on additional positional uncertainty. In contrast, the NEC only requires feature positions. While the positional uncertainty allows our PNEC to achieve more accurate rotation estimation, the information needs to be sufficiently accurate to provide a benefit. The influence of insufficient information can be seen in the results on seq. 01 of the KITTI dataset, where the KLT tracker produces wrong feature positions and uncertainty. The the poor performance of the NEC cannot be overcome by the PNEC. The PNEC puts emphasis on accurately tracked feature correspondences during its optimization. However, given poor feature correspondences it can emphasize wrong features. Therefore, outlier removal is a necessary step for the PNEC.

In our formulation of the PNEC we assumed a Gaussian error model for the feature position. This assumption is not accurate for most feature position errors. A more sophisticated error model could lead to even more accurate rotation estimates. However, it would also lead to a more intricate energy function. We consider this an interesting direction for further investigations. The performance of the PNEC on the KITTI dataset shows that the Gaussian assumption can deliver good performance for real-world data.

C. Extracting Feature Position Uncertainties from KLT Tracks

In the following, we explain how to obtain position uncertainties from the energy function. To this end, we first introduce the relationship between an energy function and the Boltzmann distribution, then the Laplace approximation, and finally apply both to Kanade-Lucas-Tomasi (KLT) tracks.

Given an energy function $E(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$ we can derive an associated probability distribution

$$p(\boldsymbol{x}) = Z^{-1} \exp(-E(\boldsymbol{x})) \tag{1}$$

that is often referred to as the *Boltzmann distribution* [3,

* equal contribution. Project page: https://go.vision.in.tum.de/pnec

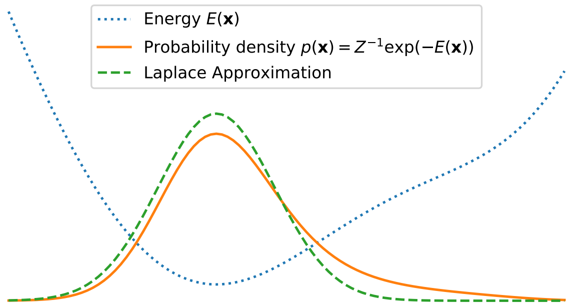


Figure 1. The plot shows the energy function $E(\boldsymbol{x})$ (*dotted blue*), the normalized distribution $p(\boldsymbol{x})$ (*orange*), together with the Laplace approximation centered on the mode $\boldsymbol{x}^*$ of $p(\boldsymbol{x})$ (*dashed green*).

Eqn. (8.41)]. The constant $Z > 0$ ensures that the distribution is normalized. If $\boldsymbol{x}^*$ is a local minimizer of $E(\boldsymbol{x})$ and thus at a local maximum of $p(\boldsymbol{x})$ we can use the *Laplace approximation* to derive a Gaussian distribution that locally approximates $p(\boldsymbol{x})$ around $\boldsymbol{x}^*$ [3, Sec. 4.4]. For an illustration please see Fig. 1. Specifically, the Gaussian distribution has mean $\boldsymbol{\mu} = \boldsymbol{x}^*$ and inverse covariance

$$\boldsymbol{\Sigma}^{-1} = \frac{d^2}{d\boldsymbol{x}^2} E(\boldsymbol{x}) \bigg|_{\boldsymbol{x}=\boldsymbol{x}^*} , \tag{2}$$

where $\frac{d^2}{d\boldsymbol{x}^2}$ denotes the Hessian matrix.

For KLT tracking we use the sum of squared normalized differences as the energy function

$$E_{KLT}(\boldsymbol{T}) = \sum_{\boldsymbol{p}\in P} \left(\frac{I_h(\boldsymbol{p})}{\bar{I}_h} - \frac{I_t(\boldsymbol{T}(\boldsymbol{p}))}{\bar{I}_t} \right)^2 , \tag{3}$$

where

$$\bar{I}_h = \frac{1}{|P|} \sum_{\boldsymbol{p}_i\in P} I_h(\boldsymbol{p}_i), \quad \bar{I}_t = \frac{1}{|P|} \sum_{\boldsymbol{p}_i\in P} I_t(\boldsymbol{T}(\boldsymbol{p}_i))$$

are the mean intensity in the host frame I_h and target frame I_t, respectively. Furthermore, $\boldsymbol{T}$ is the transformation between host and target frame that is being optimized, P is the pattern of pixels over which the sum is computed, and $|P|$ denotes the number of pixels in P.

The KLT tracking implementation from [10] that we use for our experiments on KITTI uses the *inverse compositional formulation* [1] that allows for more efficient tracking. Due to this formulation, the tracking optimizes a proxy for Eq. 3 and thus the approximation of the Hessian of the energy function is computed once in the host frame. The full explanation of the inverse compositional formulation is outside the scope of this paper and for further details we kindly refer the reader to the excellent paper by Baker and Matthews [1], which is the first in a multi-paper series.

Because the approximate Hessian is computed in the host frame only we omit the subscript and denote the host frame by I in the following. We first compute the Jacobian for each pixel $\boldsymbol{p}_i$ in the pattern P and then accumulate all Jacobians into the Gauss-Newton approximation of the Hessian. We denote the image gradient by $\nabla I(\boldsymbol{p})$ and the Jacobian w.r.t. the pixel position by $\boldsymbol{J}_{\boldsymbol{p}_i}$, which gives

$$\boldsymbol{J}_{\xi,i} = \begin{pmatrix} 1 & 0 & -p_{i,y} \\ 0 & 1 & p_{i,x} \end{pmatrix} \tag{4}$$

$$\boldsymbol{J}_{\boldsymbol{p}_i} = |P| \frac{\nabla I(\boldsymbol{p}_i)^\top \boldsymbol{J}_{\xi,i} \sum_{\boldsymbol{p}_j\in P} I(\boldsymbol{p}_j) - I(\boldsymbol{p}_i) \sum_{\boldsymbol{p}_j\in P} \nabla I(\boldsymbol{p}_j)^\top \boldsymbol{J}_{\xi,j}}{\left(\sum_{\boldsymbol{p}_j\in P} I(\boldsymbol{p}_j) \right)^2} \tag{5}$$

$$\boldsymbol{J}_{SE(2)} = \begin{pmatrix} \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \\ \vdots \\ \boldsymbol{J}_n \end{pmatrix} \tag{6}$$

$$\boldsymbol{H}_{\mathrm{SE}(2)} = \boldsymbol{J}_{SE(2)}^\top \boldsymbol{J}_{SE(2)} \tag{7}$$

$$\boldsymbol{\Sigma}_{\mathrm{SE}(2)} = \boldsymbol{H}_{\mathrm{SE}(2)}^{-1} \tag{8}$$

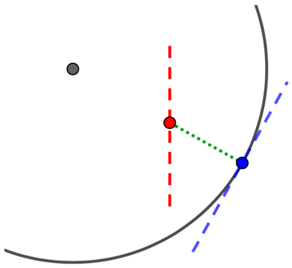
The patches are tracked w.r.t. an SE(2) transforms and thus the covariance $\boldsymbol{\Sigma}_{\mathrm{SE}(2)}$ is for the full SE(2) transform, i.e. the translation u, v in pixel coordinates as well as the 2D rotation by an angle θ. As we only consider the position, we take the marginal over the first two coordinates by selecting the upper left 2×2 sub-matrix $\boldsymbol{\Sigma}_{2\mathrm{D},h}$ in the host frame. We then transform this matrix to the target frame using the estimated rotation

$$\boldsymbol{\Sigma}_{2\mathrm{D},t} = \boldsymbol{R}_\theta \boldsymbol{\Sigma}_{2\mathrm{D},h} \boldsymbol{R}_\theta^\top , \tag{9}$$

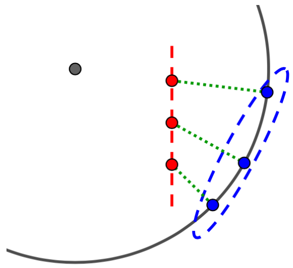
where $\boldsymbol{R}_\theta$ is the 2D rotation matrix corresponding to a rotation by an angle θ. Finally, the matrix $\boldsymbol{\Sigma}_{2\mathrm{D},t}$ is the matrix we use for the PNEC and denote $\boldsymbol{\Sigma}_{2\mathrm{D}}$ in the main paper.

D. Unscented Transform

In the following, we give an overview over the unscented transform [9] and present how it it used in the PNEC. For an illustration of the unscented transform see Fig. 2. The unscented transform gives an approximation for the mean and covariance if a non-linear transformation is applied to a Gaussian distribution. The unscented transform computes the mean and covariance from selected points to which the non-linear transformation is applied. Given a mean $\boldsymbol{\mu} \in \mathbb{R}^n$



(a) Linear approximation



(b) Unscented transform

Figure 2. Illustration of the difference between linear approximation (a) and unscented transform (b) of the projection of the covariance onto the unit-sphere in 2D. The linear approximation of the projection results in a covariance tangential to the unit-sphere—the covariance matrix does not have full rank. The unscented transform captures the non-linearity of the projection—the covariance matrix has full rank.

and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ the unscented transform selects $2n+1$ points

$$\begin{aligned} \boldsymbol{\xi}_0 &= \boldsymbol{\mu}\,, \\ w_0 &= \frac{\kappa}{n+\kappa}\,, \\ \boldsymbol{\xi}_{i,i+n} &= \boldsymbol{\mu} \pm \sqrt{n+\kappa}\boldsymbol{C}_i \quad & i = 1\dots n\,, \\ w_{i,i+n} &= \frac{1}{2(n+\kappa)} & i = 1\dots n\,, \end{aligned} \tag{10}$$

with corresponding weights around the mean, where $\boldsymbol{C}_i$ is the i-th column of the matrix $\boldsymbol{C}$ such that $\boldsymbol{\Sigma} = \boldsymbol{C}\boldsymbol{C}^\top$, and κ controls the spread of the points, which we set to the default value $\kappa = 1$. A popular way to compute $\boldsymbol{C}$ is using the Cholesky-decomposition of $\boldsymbol{\Sigma}$. The non-linear function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ is applied to the points

$$\boldsymbol{\zeta}_i = f(\boldsymbol{\xi}_i) \tag{11}$$

giving $2n+1$ points from which the new mean and covariance are computed

$$\begin{aligned} \boldsymbol{\mu}_f &= \sum_{i=0}^{2n} w_i \boldsymbol{\zeta}_i\,, \\ \boldsymbol{\Sigma}_f &= \sum_{i=0}^{2n} w_i (\boldsymbol{\zeta}_i - \boldsymbol{\mu}_f)(\boldsymbol{\zeta}_i - \boldsymbol{\mu}_f)^\top\,. \end{aligned} \tag{12}$$

For the PNEC we project the 2D covariance $\boldsymbol{\Sigma}_{\text{2D}}$ feature in the image plane onto the unit-sphere in 3D.

Pinhole Cameras. The unscented transform for pinhole cameras is straight forward. We have the feature uncertainty matrix in the image plane $\boldsymbol{\Sigma}_{\text{2D}}$ and can easily compute its Cholesky-decomposition. We sample 5 points around the feature position in the image and project them through the non-linear function $f(\boldsymbol{\xi})$. The function $f(\boldsymbol{\xi}) = h(g(\boldsymbol{\xi}))$ consist of the unprojection function

$$g(\boldsymbol{\xi}) = \boldsymbol{K}_{inv} \begin{pmatrix} \xi_1 \\ \xi_2 \\ 1 \end{pmatrix} \tag{13}$$

of the pinhole camera with the inverse camera matrix $\boldsymbol{K}_{inv}$ and the projection onto the unit sphere

$$h(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}\,. \tag{14}$$

Omnidirectional Cameras. The unscented transform for omnidirectional cameras is a bit more involved. For omnidirectional cameras image points are not on a plane but a sphere. The covariance matrix $\boldsymbol{\Sigma}_{spherical}$ corresponding to a point are therefore tangential to this sphere. They are living in a 2D subspace in a 3D space, the 3×3 covariance matrices are not positive definite and therefore the Cholesky-decomposition is not defined for them. Instead, we use the Cholesky-decomposition of a 2×2 sub-matrix $\boldsymbol{C}\boldsymbol{C}^\top = \boldsymbol{\Sigma}_{\text{2D}}$ of the form

$$\boldsymbol{R}^\top \begin{pmatrix} \boldsymbol{\Sigma}_{\text{2D}} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix} \boldsymbol{R} = \boldsymbol{\Sigma}_{spherical}\,. \tag{15}$$

A matrix $\boldsymbol{R}$ that gives us such a form is the rotation matrix

$$\boldsymbol{R} = \frac{1}{\|\boldsymbol{\mu}\|} \begin{pmatrix} \|\boldsymbol{\mu}\| - \frac{\mu_1^2}{\|\boldsymbol{\mu}\|+\mu_3} & -\frac{\mu_1\mu_2}{1+\mu_3} & -\mu_1 \\ -\frac{\mu_1\mu_2}{1+\mu_3} & \|\boldsymbol{\mu}\| - \frac{\mu_2^2}{\|\boldsymbol{\mu}\|+\mu_3} & -\mu_2 \\ \mu_1 & \mu_2 & \mu_3 \end{pmatrix}, \tag{16}$$

that aligns the feature point with the z-axis and the covariance matrix with the xy-plane, where μ_i denotes the ith element of $\boldsymbol{\mu}$.

The 5 points for the unscented transform are selected using

$$\begin{aligned} \boldsymbol{\xi}_0 &= \boldsymbol{\mu}\,, \\ \boldsymbol{\xi}_{i,i+n} &= \boldsymbol{\mu} \pm \sqrt{n+\kappa}\boldsymbol{R}^\top \begin{pmatrix} \boldsymbol{C}_i \\ 0 \end{pmatrix} \quad i = 1\dots n\,. \end{aligned} \tag{17}$$

Unlike the pinhole camera we do not need a unprojection function and the non-linear function is given by

$$f(\boldsymbol{\xi}) = \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|}\,. \tag{18}$$

E. Geometric Interpretations of the PNEC

As we explain in the main paper, the PNEC incoorperates uncertainty into the NEC and leads to a Mahalanobis-distance-based energy function. In this section, we take an in-depth look into the derivation of the energy function as a

Mahalanobis distance and give a geometric reasoning why the regularization removes the singularity of the PNEC.

In the main paper, we look at the univariate distribution of an error residual with its variance σ_i^2 that leads to the PNEC energy function. Another way to derive the same energy function is by looking at the distribution of an *epipolar plane normal vector* $\boldsymbol{n}_i$. It has a Gaussian distribution with the covariance $\boldsymbol{\Sigma}_{n,i}$. Using the formulation of the Mahalanobis distance to a plane by Schindler and Bischof [8] we can derive the distance of this normal vector to the *epipolar normal plane*. The idea of Schindler and Bischof is to apply a whitening transform to the error distribution and the plane and compute the Euclidean distance of the new center to the transformed plane.

For the PNEC we have the *epipolar normal plane* described in homogeneous coordinates by

$$\boldsymbol{p} = \begin{pmatrix} \boldsymbol{t} \\ 0 \end{pmatrix} \tag{19}$$

and the covariance with its singular value decomposition

$$\boldsymbol{\Sigma}_{n,i} = \boldsymbol{R}_i^\top \boldsymbol{V}_i \boldsymbol{R}_i, \quad \boldsymbol{V}_i = diag(a^2, b^2, c^2) \tag{20}$$

from which we can derive the whitening transform. For the plane it is given by

$$\boldsymbol{q} = \begin{pmatrix} \boldsymbol{V}_i^{1/2}\boldsymbol{R}_i & \boldsymbol{0} \\ -\boldsymbol{n}_i^\top & 1 \end{pmatrix} \boldsymbol{p} = \begin{pmatrix} \boldsymbol{V}_i^{1/2}\boldsymbol{R}_i\boldsymbol{t} \\ -\boldsymbol{n}_i^\top \boldsymbol{t} \end{pmatrix}. \tag{21}$$

The original Mahalanobis distance is now given by the distance of the origin to the transformed plane

$$d_M = \frac{q_4}{\sqrt{q_1^2 + q_2^2 + q_3^2}} \tag{22}$$

giving us

$$d_M^2 = \frac{|\boldsymbol{n}_i^\top \boldsymbol{t}|^2}{\boldsymbol{t}^\top \boldsymbol{R}_i^\top \boldsymbol{V}_i^{1/2} \boldsymbol{V}_i^{1/2} \boldsymbol{R}_i \boldsymbol{t}} = \frac{|\boldsymbol{n}_i^\top \boldsymbol{t}|^2}{\boldsymbol{t}^\top \boldsymbol{\Sigma}_{n,i} \boldsymbol{t}} \tag{23}$$

for the PNEC, which is the residual of the energy function.

Using the geometric interpretation of the PNEC as a Mahalanobis distance of the normal vector to the *epipolar normal plane* we can now explain the singularity of the PNEC geometrically. Since $\boldsymbol{n}_i = \boldsymbol{f}_i \times \boldsymbol{R}\boldsymbol{f}_i'$ is derived as a cross product its distribution is only a 2D plane embedded in 3D space. The 2D plane is described by its normal vector $\boldsymbol{f}_i$ (note that $\boldsymbol{f}_i$ is not random while $\boldsymbol{f}_i'$ is random). Therefore, the Mahalanobis distance is only defined for points in this 2D subspace. In all configuration except for $\boldsymbol{f}_i = \boldsymbol{t}$ the 2D plane and the *epipolar normal plane* intersect in a single line giving a meaningful Mahalanobis distance. The proposed regularization is equivalent to removing the 2D subspace constraint on the Mahalanobis distance by giving the covariance matrix full rank due to

$$\begin{aligned} \boldsymbol{t}^\top \left(\boldsymbol{\Sigma}_{n,i} + c\boldsymbol{I}_3\right) \boldsymbol{t} &= \boldsymbol{t}^\top \boldsymbol{\Sigma}_{n,i} \boldsymbol{t} + c\boldsymbol{t}^\top \boldsymbol{I}_3 \boldsymbol{t} \\ &= \boldsymbol{t}^\top \boldsymbol{\Sigma}_{n,i} \boldsymbol{t} + c\,. \end{aligned} \tag{24}$$

F. Directional Limit at the Singularity

In this section, we further investigate the singularity of the PNEC. Since no limit for the singularity exists, we present its directional limit. To this end, we use spherical coordinates to approach the singularity on the unit-sphere. For convenience we restate the PNEC weighted residual

$$e_{P,i}^2(\boldsymbol{R}, \boldsymbol{t}) = \frac{e_i^2}{\sigma_i^2} = \frac{|\boldsymbol{t}^\top (\boldsymbol{f}_i \times \boldsymbol{R}\boldsymbol{f}_i')|^2}{\boldsymbol{t}^\top \hat{\boldsymbol{f}}_i \boldsymbol{R} \boldsymbol{\Sigma}_i \boldsymbol{R}^\top \hat{\boldsymbol{f}}_i^\top \boldsymbol{t}}\,. \tag{25}$$

Since the limit

$$\lim_{\boldsymbol{f}_i \to \boldsymbol{t}} e_{P,i}^2(\boldsymbol{R}, \boldsymbol{t}) \tag{26}$$

does not exist, we look at the directional limit of the singularity. Without loss of generality we choose

$$\boldsymbol{t} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{27}$$

We can now approach the translation on the unit sphere by choosing an arbitrary vector

$$\boldsymbol{f}_i(\theta, \phi) = \begin{pmatrix} \sin\theta\sin\phi \\ -\sin\theta\cos\phi \\ \cos\theta \end{pmatrix} \tag{28}$$

on the unit sphere in spherical coordinates with radius 1. Letting $\theta \to 0$ implies $\boldsymbol{f}_i \to \boldsymbol{t}$. We can rewrite the residual as

$$e_{P,i}^2(\boldsymbol{R}, \boldsymbol{t}) = \frac{|(\boldsymbol{t} \times \boldsymbol{f}_i)^\top \boldsymbol{R}\boldsymbol{f}_i'|^2}{(\boldsymbol{t} \times \boldsymbol{f}_i)^\top \boldsymbol{R}\boldsymbol{\Sigma}_i \boldsymbol{R}^\top (\boldsymbol{t} \times \boldsymbol{f}_i)} \tag{29}$$

and the directional limit is given by

$$\lim_{\theta \to 0} \frac{|(\boldsymbol{t} \times \boldsymbol{f}_i(\theta))^\top \boldsymbol{R}\boldsymbol{f}_i'|^2}{(\boldsymbol{t} \times \boldsymbol{f}_i(\theta))^\top \boldsymbol{R}\boldsymbol{\Sigma}_i \boldsymbol{R}^\top (\boldsymbol{t} \times \boldsymbol{f}_i(\theta))}\,. \tag{30}$$

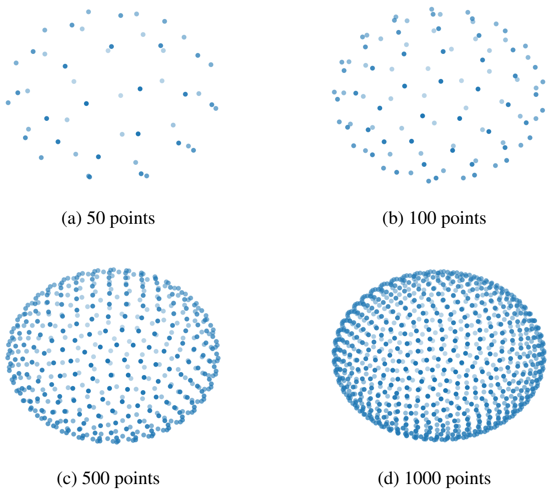
The cross product is given by

$$\boldsymbol{t} \times \boldsymbol{f}_i(\theta) = -\sin\theta \begin{pmatrix} \cos\phi \\ \sin\phi \\ 0 \end{pmatrix} = -\sin\theta \boldsymbol{k} \tag{31}$$

with $\boldsymbol{k}$ being the unit length vector orthogonal to $\boldsymbol{f}_i$ and $\boldsymbol{t}$.

$$\begin{aligned} &\lim_{\theta \to 0} \frac{|(\boldsymbol{t} \times \boldsymbol{f}_i(\theta))^\top \boldsymbol{R}\boldsymbol{f}_i'|^2}{(\boldsymbol{t} \times \boldsymbol{f}_i(\theta))^\top \boldsymbol{R}\boldsymbol{\Sigma}_i \boldsymbol{R}^\top (\boldsymbol{t} \times \boldsymbol{f}_i(\theta))} \\ &= \lim_{\theta \to 0} \frac{\sin^2\theta}{\sin^2\theta} \frac{|\boldsymbol{k}^\top \boldsymbol{R}\boldsymbol{f}_i'|^2}{\boldsymbol{k}^\top \boldsymbol{R}\boldsymbol{\Sigma}_i \boldsymbol{R}^\top \boldsymbol{k}} \\ &= \frac{|\boldsymbol{k}^\top \boldsymbol{R}\boldsymbol{f}_i'|^2}{\boldsymbol{k}^\top \boldsymbol{R}\boldsymbol{\Sigma}_i \boldsymbol{R}^\top \boldsymbol{k}} \end{aligned} \tag{32}$$

From the above equation we can clearly see that the directional limit for $\theta \to 0$ exists and depends on the direction $\boldsymbol{k}$. Consequentially, the limit in Eq. 26 does not exist.



(a) 50 points (b) 100 points

(c) 500 points (d) 1000 points

Figure 3. Fibonacci lattice point generation for different number of points. See Alg. 2 for more details.

G. SCF

We show the SCF iteration and the proposed globalization strategy in Alg. 1. For convenience we restate the PNEC energy function

$$E_P(\boldsymbol{R}, \boldsymbol{t}) = \sum_i \frac{e_i^2}{\sigma_i^2} = \sum_i \frac{|\boldsymbol{t}^\top(\boldsymbol{f}_i \times \boldsymbol{R}\boldsymbol{f}_i')|^2}{\boldsymbol{t}^\top \hat{\boldsymbol{f}}_i \boldsymbol{R}\boldsymbol{\Sigma}_i \boldsymbol{R}^\top \hat{\boldsymbol{f}}_i{}^\top \boldsymbol{t}} \tag{33}$$

that we optimize over the translation $\boldsymbol{t}$ using the SCF iteration. The Fibonacci-lattice-based point generation on the sphere in $\mathbb{R}^3$ is given in Alg. 2.

The main step of the SCF iteration is the construction of the $\boldsymbol{E}$-matrix [2, Eqn. (2.3)], which is a 3×3 symmetric matrix in our case

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{R}, \boldsymbol{t}) &= \sum_i w_i \cdot \left(\boldsymbol{t}^\top \boldsymbol{B}_i \boldsymbol{t} \cdot \boldsymbol{A}_i - \boldsymbol{t}^\top \boldsymbol{A}_i \boldsymbol{t} \cdot \boldsymbol{B}_i\right) , \\ \boldsymbol{A}_i &= \hat{\boldsymbol{f}}_i \boldsymbol{R}\boldsymbol{f}_i' (\hat{\boldsymbol{f}}_i \boldsymbol{R}\boldsymbol{f}_i')^\top , \\ \boldsymbol{B}_i &= \hat{\boldsymbol{f}}_i \boldsymbol{R}\boldsymbol{\Sigma}_i \boldsymbol{R}^\top \hat{\boldsymbol{f}}_i{}^\top + c\boldsymbol{I}_3 , \\ w_i &= (\boldsymbol{t}^\top \boldsymbol{B}_i \boldsymbol{t})^{-2} \cdot \prod_j \boldsymbol{t}^\top \boldsymbol{B}_j \boldsymbol{t} . \end{aligned} \tag{34}$$

In the SCF iteration the eigenvector corresponding to the largest eigenvalue is computed and thus, if necessary for numerical reasons, the weights can be scaled by a constant. Specifically, we use weights of the form $w_i = (\boldsymbol{t}^\top \boldsymbol{B}_i \boldsymbol{t})^{-2}$ because the product $\prod_j \boldsymbol{t}^\top \boldsymbol{B}_j \boldsymbol{t}$ is common to all weights and leads to numerical issues. Due to the regularization, the matrices $\boldsymbol{B}_i$ are positive definite, which is required for the SCF iteration. For experiments on synthetic data that follow [7] and KITTI experiments the same regularization as for the overall PNEC optimization can be used for c. This ensures, that the matrices $\boldsymbol{B}_i$ are positive definite.

Algorithm 1: SCF Optimization w/ Globalization

Data: Fixed rotation $\tilde{\boldsymbol{R}}$

Result: Optimized translation $\boldsymbol{t}^*$

Sample the Fibonacci Lattice with K points (cf. Alg. 2)

$\{\bar{\boldsymbol{t}}_k\}_k \leftarrow \mathrm{FibonacciLattice}(K)$

Select the starting point with minimal Energy (cf. Eq. 33)

$\boldsymbol{t}_0 \leftarrow \arg\min_k E_P(\tilde{\boldsymbol{R}}, \bar{\boldsymbol{t}}_k)$

for $s \leftarrow 1$ **to** S **do**

- Construct the $\boldsymbol{E}$-matrix (cf. Eq. 34)

  $\boldsymbol{E}_s \leftarrow \boldsymbol{E}(\tilde{\boldsymbol{R}}, \bar{\boldsymbol{t}}_{s-1})$

- Eigendecompose $\boldsymbol{E}_s \in \mathbb{R}^{3\times3}$ using $\boldsymbol{E}_s = \boldsymbol{E}_s^\top$

  $\lambda_1, \lambda_2, \lambda_3, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \leftarrow \mathrm{eig}(\boldsymbol{E}_s)\ s.t.\ \lambda_1 \leq \lambda_2 \leq \lambda_3$

- Set $\boldsymbol{t}_s$ as the eigenvector with maximal eigenvalue

  $\boldsymbol{t}_s \leftarrow \boldsymbol{v}_3$

end

Algorithm 2: Fibonacci Lattice Point Generation

Data: Number of points K

Result: Points on the sphere $\{\bar{\boldsymbol{t}}_k = (x_k, y_k, z_k)\}_k$

Compute the golden ratio angle

$\phi \leftarrow \pi \cdot (3 - \sqrt{5})$

for $k \leftarrow 1$ **to** K **do**

- Compute the k^{th} y-coordinate $y_k \in [-1, 1]$

  $y_k \leftarrow 1 - 2 \cdot \frac{k-1}{K-1}$

- Compute the radius in the x-z-plane

  $r_{xz} \leftarrow \sqrt{1 - y_k^2}$

- Compute the remaining coordinates x_k, z_k for $\bar{\boldsymbol{t}}_k$

  $x_k \leftarrow r_{xz} \cdot \cos((k-1)\phi)$

  $z_k \leftarrow r_{xz} \cdot \sin((k-1)\phi)$

end

H. Hyperparameters

In the following we give an overview over the parameters we use for the simulated experiments and on the KITTI dataset [6]. We show both the parameters used in the PNEC optimization and the parameters used to generate the KLT tracks.

For PNEC we use: **Alg. 1 iterations** is the number of iterations we use in Alg. 1 of the main paper before we start the least squares refinement; **SCF iterations** is the number of iteration we run Alg. 1 presented in this supplementary material in each optimization over $\boldsymbol{t}$; **Fibonacci lattice points** is the number of points we generate using the Fibonacci lattice; **regularization** is the regularization constant we proposed to avoid the singularities of the PNEC.

For the KLT parameters the parameters are: **pattern size**

Hyperparameter	Simulated	KITTI
Alg. 1 iterations	10	10
SCF itertations	10	10
Fibonacci lattice points	500	500
regularization	10^{-10}	10^{-13}
KLT parameters		
pattern size		52
grid size		30
pyramid-levels		4
optical flow iterations		40
optical flow max recovered distance		0.04
RANSAC parameters		
iterations		5000
threshold		10^{-6}

Table 1. Parameters used for our experiments.

the pattern layout used by the KLT tracker[1]; **grid size** is the length of each square for which a track is extracted; **pyramid-levels** is the number of pyramid levels over which tracks are tracked, where the scale factor between each pyramid level is 2; **optical flow iterations** is the number of iterations tracking is done on each pyramid level; **optical flow max recovered distance** is the maximum distance a track can have from its original position after forward-backward tracking (otherwise it is discarded).

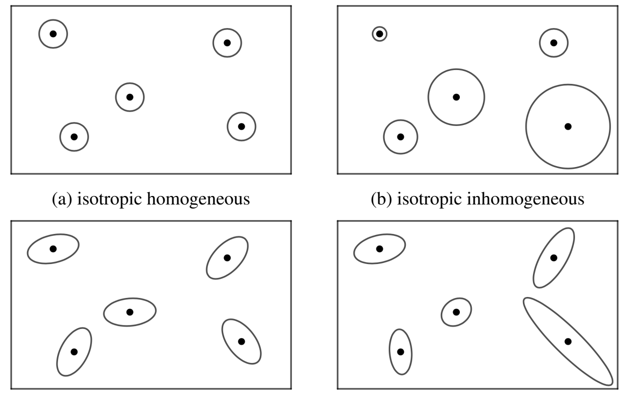
For RANSAC[2] the parameters are: **iterations** the maximum number of iterations for the RANSAC scheme; **threshold** the threshold for a point to be classified as an inlier.

H.1. Hyperparameter Study

Tab. 2 shows a the results for different hyperparameter settings on the synthetic data with anisotropic and inhomogeneous noise for different noise levels. We compare the PNEC and the NEC from the main paper and the following: sampling the Fibonacci lattice only on the first iteration of the optimization; only doing 3 iterations of the SCF algorithm after the first iteration of the optimization; sampling 5000 points with the Fibonacci lattice; increasing the regularization to $c = 10^{-5}$ and $c = 10^{-0}$. The changes to the Fibonacci lattice and the SCF algorihm lead to the same results for the PNEC. Increasing the regularization leads to worse results especially in the rotation estimation. A higher regularization leads to a more equal weighting of the residuals, the PNEC approaches the orignal NEC.

[1] see `include/basalt/optical_flow/patterns.h` in the implementation of [10]

[2] see OpenGV for details of the RANSAC scheme



(a) isotropic homogeneous (b) isotropic inhomogeneous (c) anisotropic homogeneous (d) anisotropic inhomogeneous

Figure 4. Illustration of different noise types based on Brooks *et al.* [4].

I. Experiments with Simulated Data

We present the results for the simulated experiments for other noise types. Fig. 4 shows an illustration of the noise type classification by Brooks *et al.* [4] on which we base our experiments. As for the *anisotropic inhomogonoeous* noise we present the average results for *isotropic homogonoeous*, *isotropic inhomogonoeous*, and *anisotropic homogonoeous* noise over 10 000 random instantiations.

The covariance matrices for the simulated experiments are generated using the following parameterization

$$\boldsymbol{\Sigma}_{2\mathrm{D}} = s\boldsymbol{R}_\alpha \begin{pmatrix} \beta & 0 \\ 0 & 1-\beta \end{pmatrix} \boldsymbol{R}_\alpha^\top \tag{35}$$

with a scaling factor s, an anisotropy term β, and a 2D rotation matrix

$$\boldsymbol{R}_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}. \tag{36}$$

The parameters for *isotropic homogeneous* noise are $s = 1$, $\beta = 0.5$, and $\alpha = 0$. For *isotropic inhomogeneous* noise they are $\beta = 0.5$, $\alpha = 0$, and s is sampled uniformly between 0.5 and 1.5 for each covariance. For *anisotropic homogeneous* noise $s = 1$, β is sampled uniformly between 0.5 and 1 once for each experiment, and α is sampled uniformly between 0 and π for each covariance. For *anisotropic inhomogeneous* noise all parameters are uniformly sampled for each covariance, s between 0.5 and 1.5, β between 0.5 and 1, and α between 0 and π.

I.1. Omnidirectional Cameras

Fig. 5, Fig. 6, and Fig. 7 show the results for omnidirectional cameras. The PNEC consistently achieves better results than the NEC on all noise types. Notable is that even for **isotropic homogeneous** noise the PNEC ouperforms the

	Omnidirectional									Pinhole									
	w/ T						w/o T			w/ T						w/o T			
Noise level [px]	0.5		1.0		1.5		0.5	1.0	1.5	0.5		1.0		1.5		0.5	1.0	1.5	
Metric [degree]	e_{rot}	e_{t}	e_{rot}	e_{t}	e_{rot}	e_{t}	e_{rot}	e_{rot}	e_{rot}	e_{rot}	e_{t}	e_{rot}	e_{t}	e_{rot}	e_{t}	e_{rot}	e_{rot}	e_{rot}	
PNEC	0.08	1.29	0.12	1.60	0.14	1.66	0.09	0.13	0.15	0.20	2.06	0.28	2.38	0.34	2.54	0.15	0.21	0.25	
NEC	0.11	1.90	0.15	2.10	0.17	2.11	0.11	0.15	0.18	0.25	2.41	0.34	2.78	0.41	2.91	0.19	0.25	0.29	+24 %
Fib. lattice only for 1st	0.08	1.30	0.12	1.60	0.14	1.66	0.09	0.13	0.15	0.20	2.05	0.28	2.37	0.34	2.56	0.15	0.21	0.25	0 %
Only 3 SCF iter. after 1st	0.08	1.32	0.12	1.60	0.14	1.68	0.09	0.13	0.15	0.20	2.05	0.28	2.37	0.34	2.55	0.15	0.21	0.25	0 %
Fib. lattice 5000 pts	0.08	1.29	0.12	1.60	0.14	1.66	0.09	0.13	0.15	0.20	2.06	0.28	2.38	0.34	2.54	0.15	0.21	0.25	0 %
Reg. $\uparrow$ $(c = 10^{-5})$	0.10	1.23	0.14	1.54	0.16	1.66	0.10	0.14	0.17	0.24	2.32	0.33	2.91	0.40	3.11	0.12	0.16	0.19	+6 %
Reg. $\uparrow\uparrow$ $(c = 10^{0})$	0.10	1.23	0.14	1.54	0.16	1.66	0.10	0.14	0.17	0.24	2.32	0.33	2.91	0.40	3.11	0.12	0.16	0.19	+6 %

Table 2. Hyperparameter study on the synthetic data. We compare: the PNEC and the NEC from the main paper; sampling the Fibonacci lattice only on the first iteration of the optimization; only doing 3 iterations of the SCF algorithm after the first iteration of the optimization; sampling 5000 points with the Fibonacci lattice; increasing the regularization to $c = 10^{-5}$ and $c = 10^{-0}$. The changes to the Fibonacci lattice and the SCF algorihm lead to the same results for the PNEC. Increasing the regularization leads to worse results especially in the rotation estimation. A higher regularization leads to a more equal weighting of the residuals, the PNEC approaches the orignal NEC.

NEC. This experiment shows that the geometry of the problem has an influence on the energy function. Although all covariances are equal in the image plane the same does not hold for the variance of each residual. The experiments for **isotropic inhomogeneous** and **anisotropic homogeneous** noise show the benefit the differnent sizes and shape of covariances has, respectively. Both widen the gap between the PNEC and NEC.

I.2. Pinhole Cameras

Fig. 8, Fig. 9, and Fig. 10 show the results for pinhole cameras. They show similar results to the omnidirectional cameras.

J. Energy Function Results

Fig. 11 shows the median energy values for all the simulated experiments presented in the main paper and this supplementary material. We show the median values instead of the average due to the volatility of the energy function.

K. Anisotropy Experiments

As the results of the previous experiment shows, the performance of the PNEC is dependent on the noise type. The following experiments show the effect the anisotropy of the noise has on the PNEC optimization. The setup for this experiment is the same as for the previous experiments, apart from the noise generation. While the noise level stays constant (1.0 [pix]), inhomogenous noise is sampled over different levels of anisotropy β ($\beta = 0.5$: isotropic, $\beta = 1.0$: anisotropic). Fig. 12 and Fig. 13 show the results for omnidirectional and pinhole cameras, respectively. We repeated the experiments for pure rotation. An increasing level of anisotropy is beneficial for our PNEC, increasing the gap between the NEC and the PNEC noticeably. Additionally, the results show that our PNEC outperforms the NEC even for isotropic noise.

L. Offset Experiments

The previous synthetic experiments showed the robustness of our PNEC against different noise intensities. For these experiments, we assumed perfect knowledge about the noise distribution. Fig. 14 shows the influence of wrongly estimated noise parameters on the performance of our PNEC. For these experiments we generated random pinhole camera problems but gave the PNEC covariance matrices with a random offset on the noise parameters. The offset is uniformly sampled with a deviation up to a certain percentage of the range of the noise parameters α, β, s, respectively. The results show that the PNEC performs better, the more accurate the covariance matrices are. However, our PNEC outperforms the NEC even with noise parameters that are off by up to 25%.

M. KITTI experiments

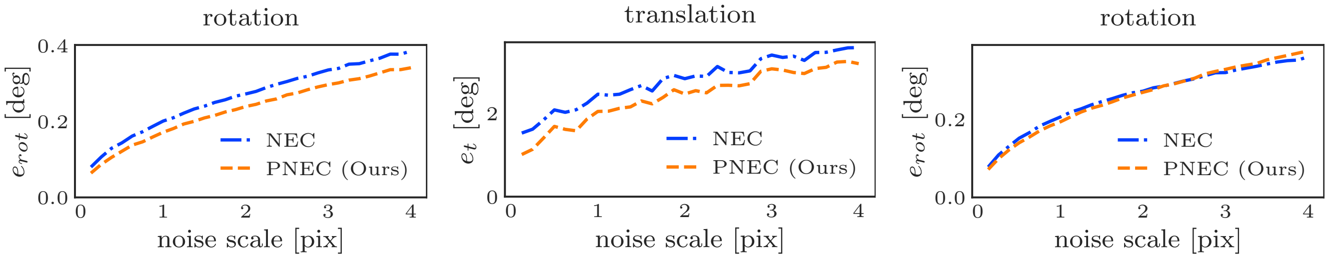
In the following we present the results on the KITTI dataset in more detail. In Sec. M.1 we compare the NEC and PNEC in their translation estimation as well as the consistency of their results. Sec. M.2 gives an ablation study on the KITTI dataset to evaluate the performance of both stages of our optimization scheme.

M.1. Translation and Variance

Fig. 15 shows the mean error and standard deviation of the KLT-NEC and the KLT-PNEC on the KITTI dataset for the RPE_1, the RPE_1 and e_{t} metrics. We chose to omit seq. 01 since neither tracks nor covariances provided by the KLT tracker are correct. The results show that our PNEC not only achieves better result on average in the rotation estimation but its results are more consistent for most sequences. The translation estimation for both methods is very similar.

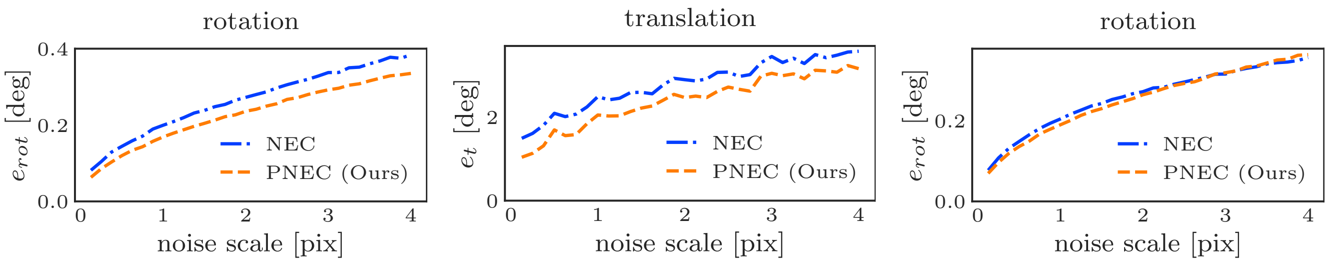
M.2. Ablation study

Tab. 3 shows the results for the ablation study on the KITTI dataset. We compare: KLT-NEC; NEC followed by the 2nd stage of our optimization scheme (KLT-NEC + PNEC-JR); only the 1st stage (KLT-PNEC w/o JR); only the 2nd stage (KLT-PNEC only JR); both stages of the optimization (KLT-PNEC). Additionally we included the MRO results as a baseline. For KITTI all methods are initialized with a constant velocity motion model. The full optimization with both stages gives the best results on most sequences and performs the best on average. Both stages on their own achieve considerably worse results. Fig. 16 shows qualitative trajectories for KLT-PNEC only JR and KLT-PNEC of five runs on seq. 07 of the KITTI dataset. While our full PNEC achieves consistent results over all runs, KLT-PNEC only JR results are considerably more volatile. The joint refinement is prone to local mimima due its least-squares formulation. This leads to a few poor rotation estimations that worsen the performance on the sequence drastically.



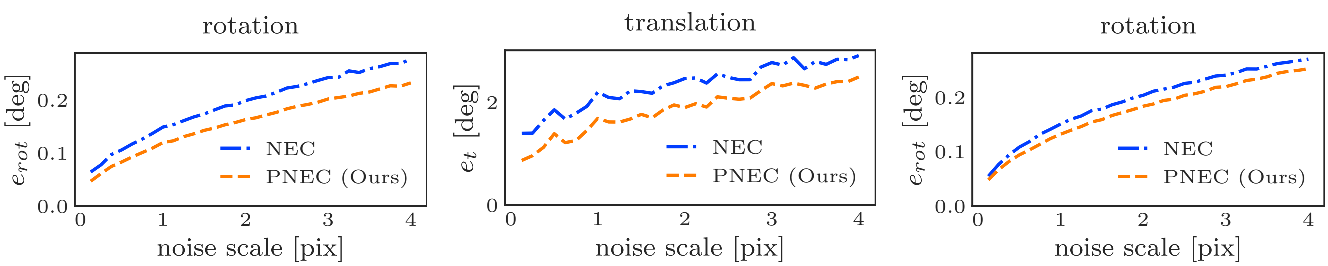
(a) Experiment w/ translation: e_{rot} (b) Experiment w/ translation: e_t (c) Experiment w/o translation: e_{rot}

Figure 5. **isotropic homogeneous**: Experiments for omnidirectional cameras. The PNEC outperforms the NEC for experiments with translation, without translation both methods perform similar. The results show that next to the shape of the covariances the geometry also influences the variance of the residuals, resulting in better rotation estimation of the PNEC.



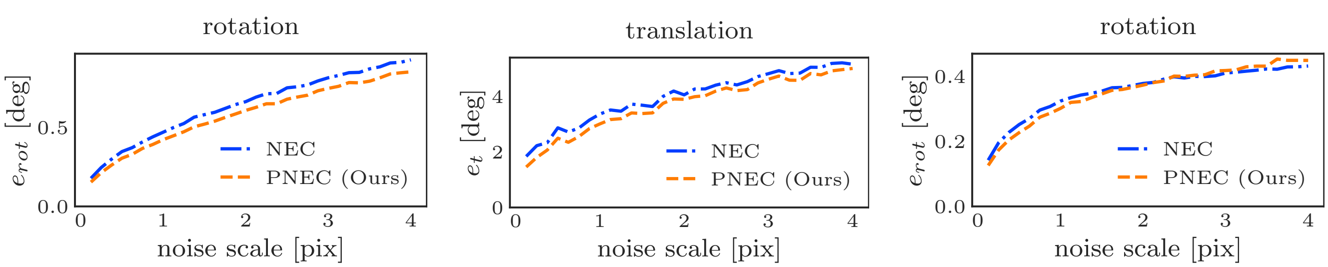
(a) Experiment w/ translation: e_{rot} (b) Experiment w/ translation: e_t (c) Experiment w/o translation: e_{rot}

Figure 6. **isotropic inhomogeneous**: Experiments for omnidirectional cameras. The PNEC outperforms the NEC for experiments with translation, without translation the PNEC only performs slightly better. As expected the inhomogeneity of the covariance matrices is beneficial for the PNEC.



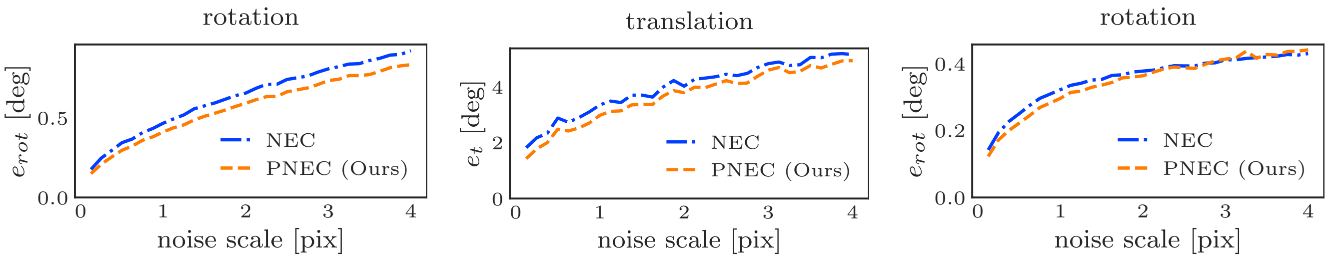
(a) Experiment w/ translation: e_{rot} (b) Experiment w/ translation: e_t (c) Experiment w/o translation: e_{rot}

Figure 7. **anisotropic homogeneous**: Experiments for omnidirectional cameras. The PNEC outperforms the NEC for experiments with translation and without translation. This experiment shows the importance the directional information of the anisotropic covariances provides the PNEC. The performance difference is significantly bigger than for isotropic noise.



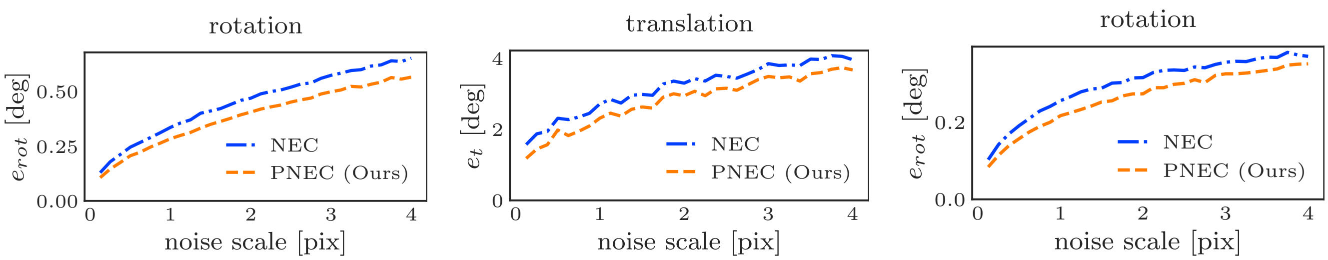
(a) Experiment w/ translation: e_{rot} (b) Experiment w/ translation: e_t (c) Experiment w/o translation: e_{rot}

Figure 8. **isotropic homogeneous**: Experiments for pinhole cameras. Similar to omnidirectional cameras the PNEC outperforms the NEC even for isotropic homogeneous noise. However, for purely rotational experiments the PNEC performs slightly worse for experiments with a high noise level.



(a) Experiment w/ translation: e_{rot} (b) Experiment w/ translation: e_t (c) Experiment w/o translation: e_{rot}

Figure 9. **isotropic inhomogeneous**: Experiments for pinhole cameras. Similar to omnidirectional cameras the inhomogeneity helps the PNEC. However, for purely rotational experiments both methods performs similar for experiments with a high noise level.



(a) Experiment w/ translation: e_{rot} (b) Experiment w/ translation: e_t (c) Experiment w/o translation: e_{rot}

Figure 10. **anisotropic homogeneous**: Experiments for pinhole cameras. Similar to omnidirectional cameras the directional information of the anisotropic covariances helps the PNEC significantly, especially in the zero translation case.

	MRO [5]		KLT-NEC		KLT-NEC + PNEC-JR		KLT-PNEC w/o JR		KLT-PNEC only JR		KLT-PNEC (Ours)	
Seq.	RPE_1	RPE_n	RPE_1	RPE_n	RPE_1	RPE_n	RPE_1	RPE_n	RPE_1	RPE_n	RPE_1	RPE_n
00	0.360	8.670	0.125	5.922	0.139	5.392	0.142	9.594	0.210	14.929	**0.119**	**3.429**
02	0.290	16.030	0.093	6.693	0.111	8.781	0.100	10.259	**0.077**	**5.271**	0.122	9.687
03	0.280	5.470	0.073	2.728	0.063	1.717	0.065	2.022	**0.057**	1.464	0.059	**1.411**
04	0.040	1.080	0.041	0.619	**0.038**	0.477	**0.038**	0.959	**0.038**	**0.448**	**0.038**	0.463
05	0.250	11.360	0.079	4.489	0.082	3.659	0.094	6.313	0.078	6.563	**0.070**	**3.203**
06	0.180	4.720	0.073	3.162	0.072	2.988	0.054	**1.577**	0.043	2.366	**0.042**	2.322
07	0.280	7.490	0.105	4.640	0.096	2.092	0.135	7.592	0.545	25.381	**0.074**	**2.065**
08	0.270	9.210	0.070	5.523	0.084	5.004	0.072	5.973	0.127	27.343	**0.060**	**3.347**
09	0.280	9.850	0.088	3.533	0.088	**2.699**	**0.079**	2.783	0.202	23.842	0.080	3.514
10	0.380	13.250	0.073	**3.959**	0.075	4.155	0.077	4.679	0.177	15.778	**0.072**	4.094
mean	0.261	8.713	0.082	4.127	0.085	3.696	0.086	5.175	0.155	12.338	**0.074**	**3.354**

Table 3. Ablation study on the KITTI dataset. We compare: KLT-NEC; NEC followed by the 2nd stage of our optimization scheme (KLT-NEC + PNEC-JR); only the 1st stage (KLT-PNEC w/o JR); only the 2nd stage (KLT-PNEC only JR); both stages of the optimization (KLT-PNEC). Additionally we included the MRO results as a baseline. For KITTI all methods are initialized with a constant velocity motion model, i.e. with the relative rotation generated for the previous frame pair by the same method. The results show that only both stages of our optimization achieve the most consistent and best results on average. Using only one stage performs worse. Especially the 2nd stage struggles on most sequences. While it has the best results on some sequences on others is performs the worst often by a wide margin in the RPE_n metric. The joint refinement is dependent on a good initialization, which is provided by the 1st stage of our optimization. KLT-NEC + PNEC-JR shows that using the NEC is not enough to provide the best results.

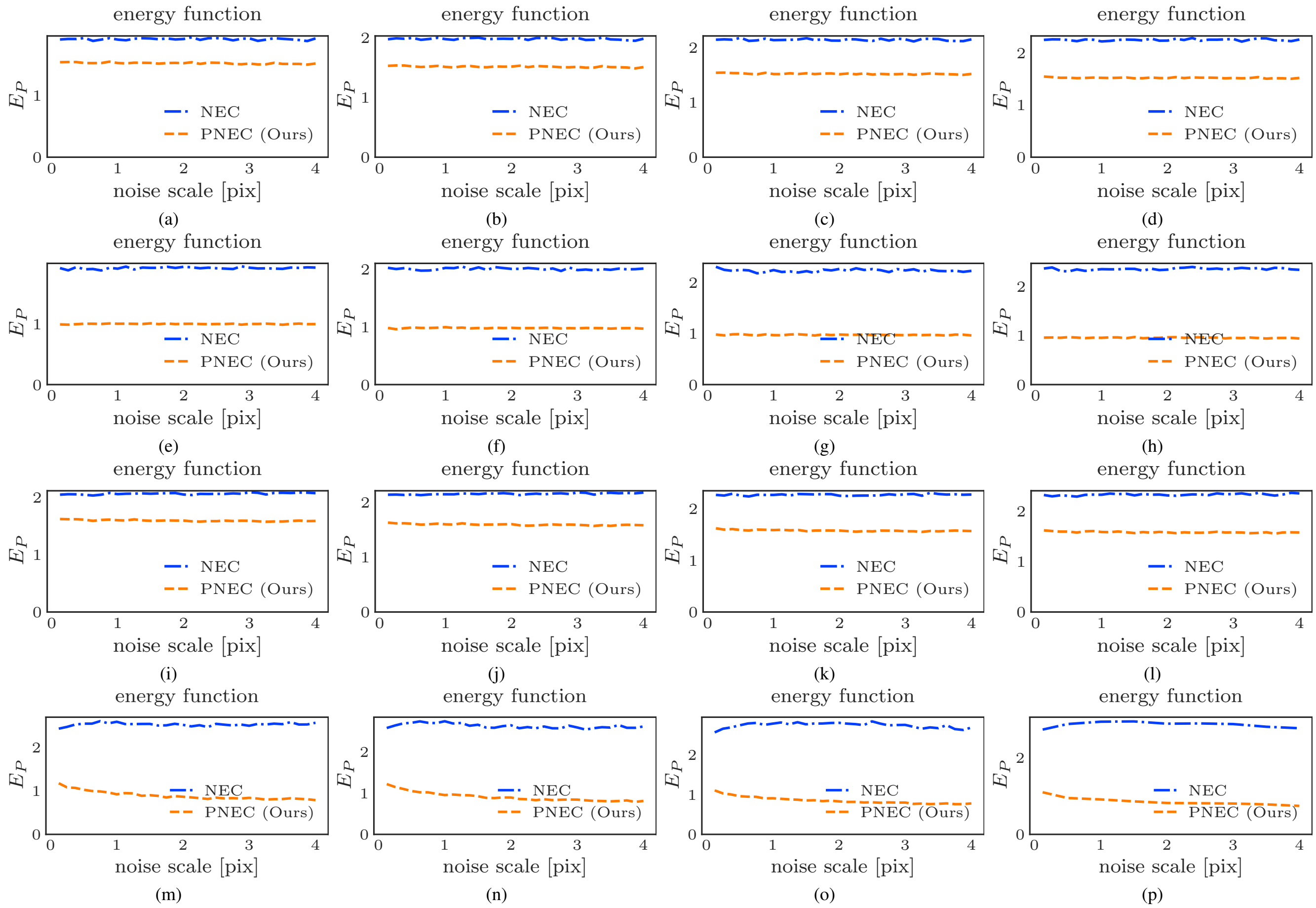
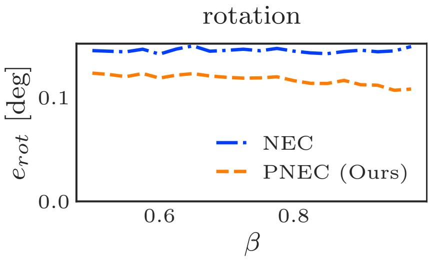
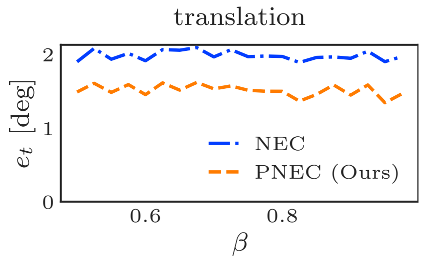


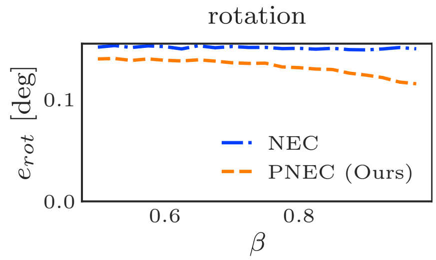
Figure 11. The energy function values for all simulated experiments. We present the median instead of the average value for the energy function, due to its volatility. The first column shows the experiments for *isotropic homogeneous* noise, the second for *isotropic inhomogeneous* noise, the third for *anisotropic homogeneous* noise, and the fourth for *anisotropic inhomogeneous* noise. The first two rows show the experiments for omnidirectional cameras with and without translation, respectively. The last two rows show the experiments for pinhole cameras with and without translation, respectively. The results show the effectiveness of our proposed optimization scheme to minimize the energy function.



(a) Experiment w/ translation: e_{rot}

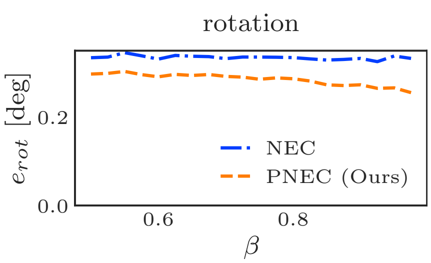


(b) Experiment w/ translation: e_t

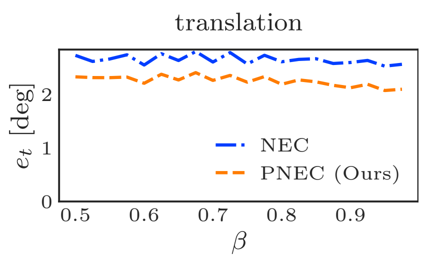


(c) Experiment w/o translation: e_{rot}

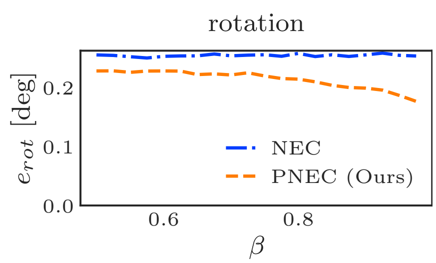
Figure 12. Influence of the degree of anisotropy for omnidirectional cameras. Results averaged of 10 000 random problems with varying degree of anisotropy. While a higher anisotropy is beneficial for our PNEC it outperforms the NEC for isotropic noise ($\beta = 0.5$) as well.



(a) Experiment w/ translation: e_{rot}

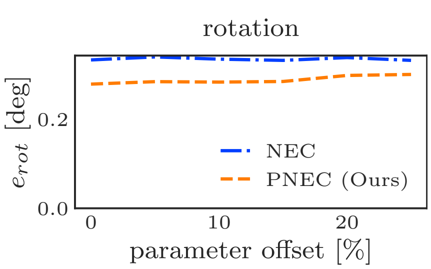


(b) Experiment w/ translation: e_t

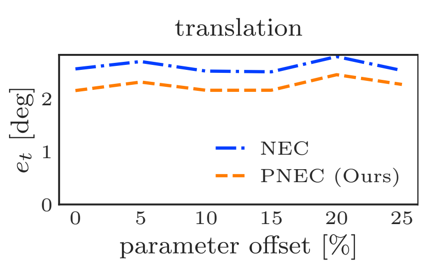


(c) Experiment w/o translation: e_{rot}

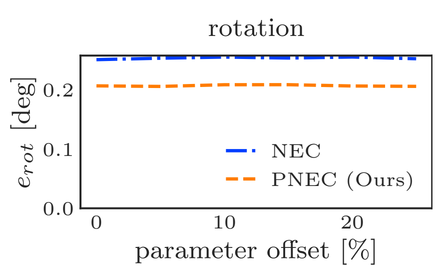
Figure 13. Influence of the degree of anisotropy for pinhole cameras. Results are similar as for omnidirectional cameras.



(a) Experiment w/ translation: e_{rot}

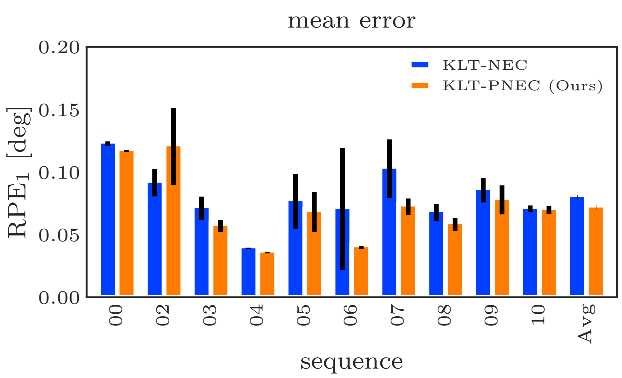


(b) Experiment w/ translation: e_t

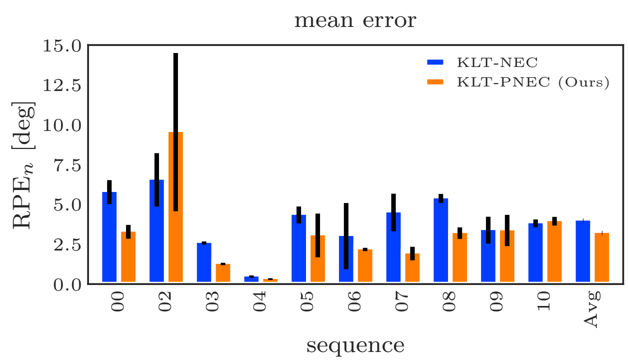


(c) Experiment w/o translation: e_{rot}

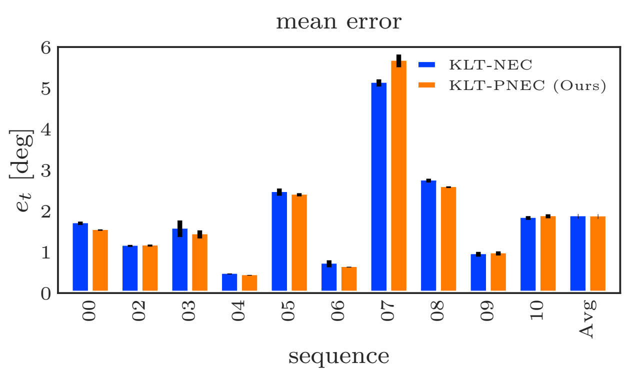
Figure 14. Influence of wrongly estimated noise parameters. Our PNEC is given covariance matrices with wrong noise parameters (see Sec. I) for the optimization. Results are averaged over 10 000 random problems over different levels of maximum parameter offset. Our PNEC outperforms the NEC even if the noise parameters have an offset of up to 25%.



(a) RPE_1



(b) RPE_n



(c) e_t

Figure 15. Mean error with standard deviation over 5 runs of the KITTI dataset. Comparison between the NEC and our PNEC shows that not only does our PNEC better results on average for both the RPE_1 (see Fig. 15a) and the RPE_n (see Fig. 15b) but the results are often more consistent. The translational error for both methods are very similar on average.

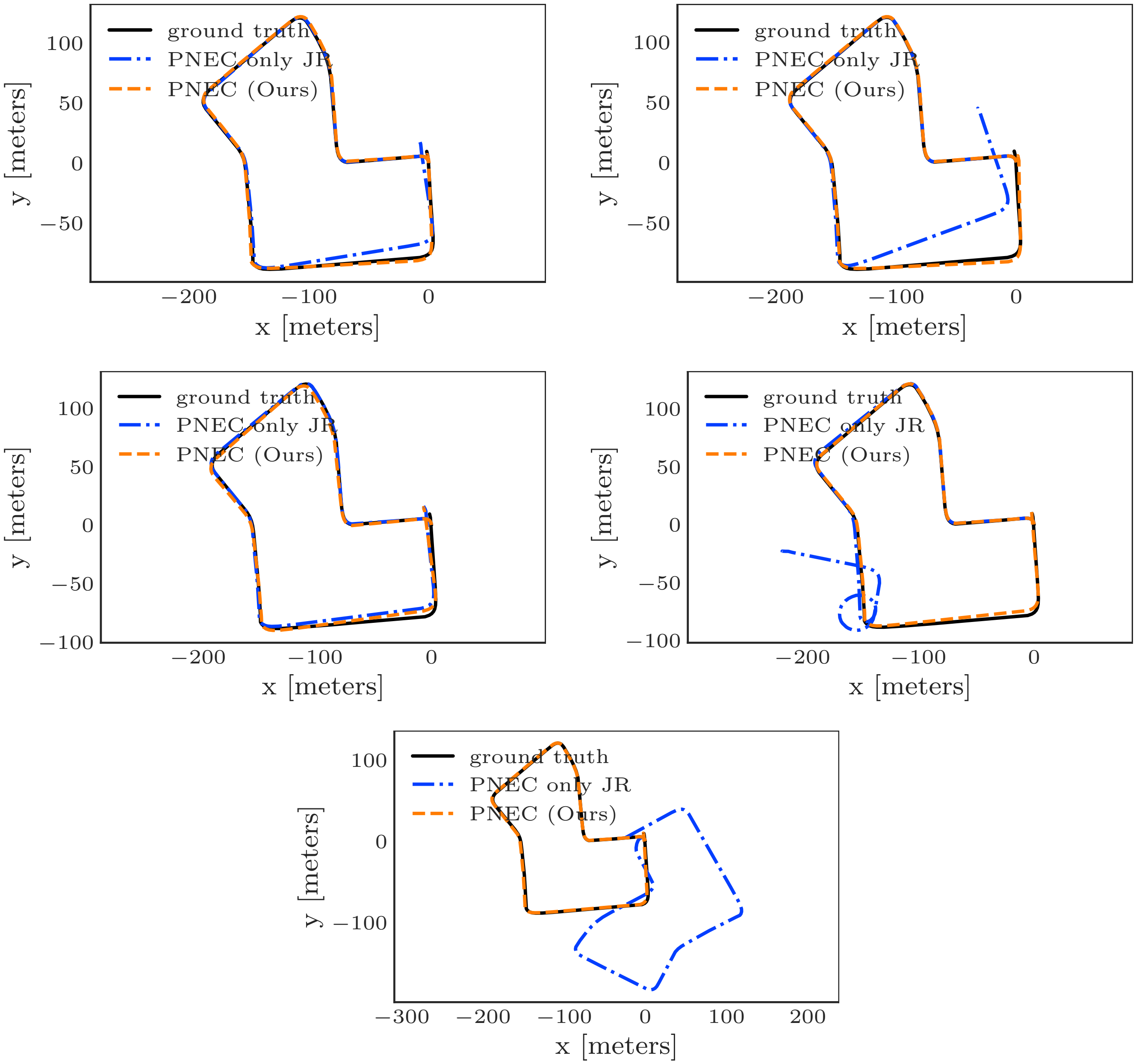


Figure 16. Qualitative evaluation for KITTI seq. 07. The trajectories are generated with the estimated rotations of PNEC only JR and our PNEC, respectively, and are combined with the ground truth translations for visualization purposes. The trajectories show the volatility of the PNEC only JR. While the rotations estimated with only the joint refinement are good over most of the sequence, it is prone to local minima due to the least-square formulation. A handful of bad rotation estimations, mainly located in corners, lead to an overall poor performance on the whole sequence. The first stage of the proposed optimization scheme overcomes this issue by providing a good initialization for the joint refinement.

References

[1] Simon Baker and Iain A. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56, 2004. 2

[2] Aohud Abdulrahman Binbuhaer. *On Optimizing the Sum of Rayleigh Quotients on the Unit Sphere*. PhD thesis, The University of Texas at Arlington, 2019. 5

[3] Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Springer, 2007. 1, 2

[4] M.J. Brooks, W. Chojnacki, D. Gawley, and A. van den Hengel. What value covariance information in estimating vision parameters? In *ICCV*, 2001. 6

[5] Chee-Kheng Chng, Álvaro Parra, Tat-Jun Chin, and Yasir Latif. Monocular rotational odometry with incremental rotation averaging and loop closure. *Digital Image Computing: Techniques and Applications (DICTA)*, 2020. 10

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 5

[7] Laurent Kneip and Simon Lynen. Direct optimization of frame-to-frame rotation. In *ICCV*, 2013. 5

[8] K. Schindler and H. Bischof. On robust regression in photogrammetric point clouds. In *DAGM-Symposium*, 2003. 4

[9] Jeffrey K Uhlmann. *Dynamic map building and localization: New theoretical foundations*. PhD thesis, University of Oxford Oxford, 1995. 2

[10] Vladyslav Usenko, Nikolaus Demmel, David Schubert, Jörg Stückler, and Daniel Cremers. Visual-inertial mapping with non-linear factor recovery. *IEEE Robotics and Automation Letters (RAL)*, 5, 2020. 2, 6