Learning Correspondence Uncertainty via Differentiable Nonlinear Least Squares

Dominik Muhle[1,2] Lukas Koestler[1,2] Krishna Murthy Jatavallabhula[4] Daniel Cremers[1,2,3]
[1]TU Munich [2]Munich Center for Machine Learning [3]University of Oxford [4]MIT
{dominik.muhle, lukas.koestler, cremers}@tum.de jkrishna@mit.edu

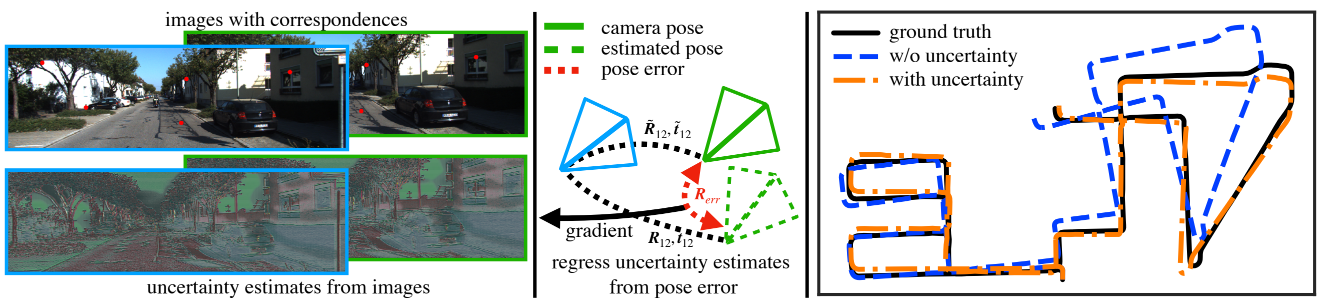


Figure 1. We present a **differentiable nonlinear least squares (DNLS) framework for learning feature correspondence quality** by computing per-feature positional uncertainty. The uncertainty estimates (left, bottom images) are regressed from a pose estimation error (middle), enabling the framework across a range of (handcrafted, learned) feature extractors. Our learned covariances (right, orange trajectory) improve orientation estimation by up to 11% over state-of-the-art probabilistic pose estimation methods on the KITTI dataset [21].

Abstract

We propose a differentiable nonlinear least squares framework to account for uncertainty in relative pose estimation from feature correspondences. Specifically, we introduce a symmetric version of the probabilistic normal epipolar constraint, and an approach to estimate the covariance of feature positions by differentiating through the camera pose estimation procedure. We evaluate our approach on synthetic, as well as the KITTI and EuRoC real-world datasets. On the synthetic dataset, we confirm that our learned covariances accurately approximate the true noise distribution. In real world experiments, we find that our approach consistently outperforms state-of-the-art non-probabilistic and probabilistic approaches, regardless of the feature extraction algorithm of choice.

1. Introduction

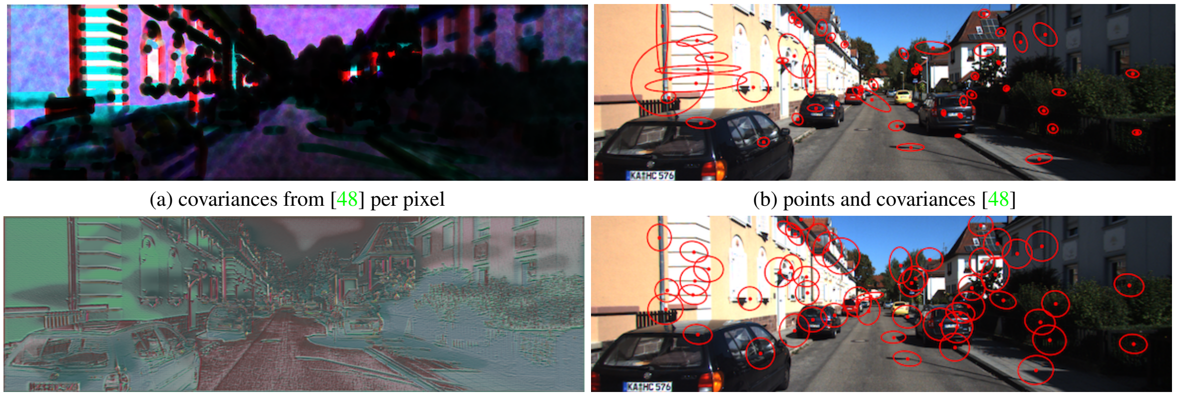
Estimating the relative pose between two images given mutual feature correspondences is a fundamental problem in computer vision. It is a key component of structure from motion (SfM) and visual odometry (VO) methods which in turn fuel a plethora of applications from autonomous vehicles or robots to augmented and virtual reality.

Estimating the relative pose – rotation and translation – between two images, is often formulated as a geometric problem that can be solved by estimating the essential matrix [42] for calibrated cameras, or the fundamental matrix [24] for uncalibrated cameras. Related algorithms like the eight-point algorithm [23, 42] provide fast solutions. However, essential matrix based approaches suffer issues such as *solution multiplicity* [18, 24] and *planar degeneracy* [33]. The normal epipolar constraint (NEC) [34] addresses the issues by estimating the rotation independently of the translation, leading to more accurate relative poses [33].

Neither of the aforementioned algorithms takes into account the *quality* of feature correspondences – an important cue that potentially improves pose estimation accuracy. Instead, feature correspondences are classified into inliers and outliers through a RANSAC scheme [11]. However, keypoint detectors [12, 56] for feature correspondences or tracking algorithms [63] yield imperfect points [40] that exhibit a richer family of error distributions, as opposed to an inlier-outlier distribution family. Algorithms, that make use of feature correspondence quality have been proposed for essential/fundamental matrix estimation [7, 53] and for the NEC [48], respectively.

While estimating the relative pose can be formulated as a classical optimization problem [15, 33], the rise in popularity of deep learning has led to several works augmenting

Project Page



(a) covariances from [48] per pixel

(b) points and covariances [48]

(c) learned covariances per pixel (Ours)

(d) points and learned covariances (Ours)

Figure 2. Comparison between covariances used in [48] (first row) and our learned covariances (second row). The first column shows a dense color coded (s, α, β mapped to HLS with γ correction) representation for each pixel, while the second column shows subsampled keypoints and their corresponding (enlarged) covariances. The higher saturation in (a) shows that the covariances are more anisotropic. The learned covariances (c) show a more fine-grained detail in the scale (brightness) and less blurring than the covariances in (a).

VO or visual simultaneous localisation and mapping (VSLAM) pipelines with learned components. GN-Net [67] learns robust feature representations for direct methods like DSO [15]. For feature based methods Superpoint [12] provides learned features, while Superglue [57] uses graph neural networks to find corresponding matches between feature points in two images. DSAC introduces a differential relaxation to RANSAC that allows gradient flow through the otherwise non-differentiable operation. In [53] a network learns to re-weight correspondences for estimating the fundamental matrix. PixLoc [58] estimates the pose from an image and a 3D model based on direct alignment.

In this work we combine the predictive power of deep learning with the precision of geometric modeling for highly accurate relative pose estimation. Estimating the noise distributions for the feature positions of different feature extractors allows us to incorporate this information into relative pose estimation. Instead of modeling the noise for each feature extractor explicitly, we present a method to learn these distributions from data, using the same domain that the feature extractors work with - images. We achieve this based on the following technical contributions:

- We introduce a symmetric version of the probabilistic normal epipolar constraint (PNEC), that more accurately models the geometry of relative pose estimation with uncertain feature positions.
- We propose a learning strategy to minimize the relative pose error by learning feature position uncertainty through differentiable nonlinear least squares (DNLS), see Fig. 1.
- We show with synthetic experiments, that using the gradient from the relative pose error leads to meaningful estimates of the positional uncertainty that reflect the correct error distribution.
- We validate our approach on real-world data in a visual odometry setting and compare our method to non-probabilistic relative pose estimation algorithms, namely Nistér 5pt [50], and NEC [33], as well as to the PNEC with non-learned covariances [48].
- We show that our method is able to generalize to different feature extraction algorithms such as SuperPoint [12] and feature tracking approaches on real-world data.
- We release the code for all experiments and the training setup to facilitate future research.

2. Related Work

This work is on deep learning for improving frame-to-frame relative pose estimation by incorporating feature position uncertainty with applications to visual odometry. We therefore restrict our discussion of related work to relative pose estimation in visual odometry, weighting correspondences for relative pose estimation, and deep learning in the context of VSLAM. For a broader overview over VSLAM we refer the reader to more topic-specific overview papers [10, 65] and to the excellent books by Hartley and Zisserman [24] and by Szeliski [62].

Relative Pose Estimation in Visual Odometry. Finding the relative pose between two images has a long history in computer vision, with the first solution for perspective images reaching back to 1913 by Kruppa [35]. Modern methods for solving this problem can be classified into *feature-based* and *direct* methods. The former rely on feature points extracted in the images together with geometric constraints like the *epipolar constraint* or the *normal epipolar constraint* [34] to calculate the relative pose. The latter optimize the pose by directly considering the intensity differences between the two images and rose to popularity with LSD-SLAM [16] and DSO [15]. Since direct methods work on the assumption of brightness or irradiance constancy they require the appearance to be somewhat similar across images. In turn, keypoint based methods rely on suitable feature extractors which can exhibit significant amounts of noise and uncertainty. In this paper we propose a method to learn the intrinsic noise of keypoint detectors – therefore, the following will focus on feature based relative pose estimation.

One of the most widely used parameterizations for reconstructing the relative pose from feature correspondences is the essential matrix, given calibrated cameras, or the fundamental matrix in the general setting. Several solutions based on the essential matrix have been proposed [36, 38, 42, 50, 61]. They include the linear solver by Longuet-Higgins [42], requiring 8 correspondences, or the solver by Nistér *et al*. [51] requiring the minimal number of 5 correspondences. However, due to their construction, essential matrix methods deteriorate for purely rotational motion with noise-free correspondences [33]. As an alternative, methods that do not use the essential matrix have been proposed – they either estimate the relative pose using quaternions [17] or make use of the normal epipolar constraint (NEC) by Kneip and Lynen [33, 34]. The latter addresses the problems of the essential matrix by estimating rotation independent of the translation. [6] shows how to obtain the global minimum for the NEC. Further work, that disentangles rotation and translation can be found in [39].

Weighting of Feature Correspondences. Keypoints in images can exhibit significant noise, deteriorating the performance for pose estimation significantly [22]. The noise characteristics of the keypoint positions depend on the feature extractor. For Kanade-Lucas-Tomasi (KLT) tracking [44, 63] approaches, the position uncertainty has been investigated in several works [20, 59, 60, 72]. The uncertainty was directly integrated into the tracking in [14]. [71] proposed a method to obtain anisotropic and inhomogeneous covariances for SIFT [43] and SURF [3].

Given the imperfect keypoint positions, not all correspondences are equally well suited for estimating the relative pose. [22] showed the effect of the noise level on the accuracy of the pose estimation. Limiting the influence of bad feature correspondences has been studied from a geometrical and a probabilistic perspective. random sample consensus (RANSAC) [19] is a popular method to classify datapoints into inliers and outliers that can be easily integrated into feature based relative pose estimation pipelines. Ranftl *et al*. [53] relax the hard classification for inlier and outlier and use deep learning to find a robust fundamental matrix estimator in the presence of outliers in an iteratively reweighted least squares (IRLS) fashion. DSAC [5] models RANSAC as a probabilistic process to make it differentiable. Other lines of work integrate information about position uncertainty directly into the alignment problem. For radar based SLAM, [8] incorporates keypoint uncertainty in radar images, with a deep network predicting the uncertainty. Image based position uncertainty was investigated from the statistical, [27, 28], the photogrammetry [46] and the computer vision perspective [7,29]. [7] and [29] debated the benefit of incorporating position uncertainty into fundamental matrix estimation. We base our method on the probabilistic normal epipolar constraint (PNEC) [48], that improved on the NEC by extending it to a probabilistic view. It achieved better results on real-world data with covariances approximated using the Boltzmann distribution [4]. We expand on this idea by learning covariances (see Fig. 2) agnostic of the keypoints extractor used to further improve pose estimation.

Deep Learning in VSLAM. Deep Learning has transformed computer vision in the last decade. While deep networks have been successfully used for tasks like detection [54], semantic segmentation [41], and recently novel view synthesis [47], they have also found application in VSLAM pipelines. DVSO [69] and D3VO [68] leveraged deep learning to improve the precision for direct methods, while GN-Net [67] predicts robust and dense feature maps. Several works proposed to learn keypoint extractors, for feature based pose estimation, such as SuperPoint [12] and LIFT [70]. SuperGlue [57] enabled feature matching with graph neural networks. Other lines of work leverage deep learning for localization by making parts of the pose estimation pipeline differentiable [2,5,58,64]. Works, that directly predicting the pose include PoseNet [30] and CTCNet [25] that uses self-supervised learning with a cycle-consistency loss for VO. [40] learns image representations by refining keypoint positions and camera poses in a post-processing step of a structure-from-motion pipeline. ∇SLAM [26] presents a differentiable dense SLAM system with several components (e.g., the Levenberg-Marquardt [37, 45] optimizer).

3. Method

In the following, we present our framework to estimate positonal uncertainty of feature points by leveraging DNLS. We learn the noise covariances through a forward and backward step. In the forward step, the covariances are used in a probabilistic pose estimation optimization, namely the PNEC. In the backward step, the gradient from the pose error is back-propagated through the optimization to the covariances. From there we can train a neural network to predict the keypoint position uncertainty from the images. We start by summarizing the asymmetric PNEC [48] and for the first time introduce its symmetric counterpart.

3.1. Prerequisites

Notation. We follow the notation of [48]. Bold lowercase letters (*e.g.* $\boldsymbol{f}$) denote vectors, whereas bold uppercase letters (*e.g.* $\boldsymbol{\Sigma}$) denote matrices. $\hat{\boldsymbol{u}} \in \mathbb{R}^{3\times3}$ represents the skew-symmetric matrix of the vector $\boldsymbol{u} \in \mathbb{R}^3$ such that the cross product between two vectors can be rewritten as a matrix-vector operation, i.e. $\boldsymbol{u} \times \boldsymbol{v} = \hat{\boldsymbol{u}}\boldsymbol{v}$. The transpose is

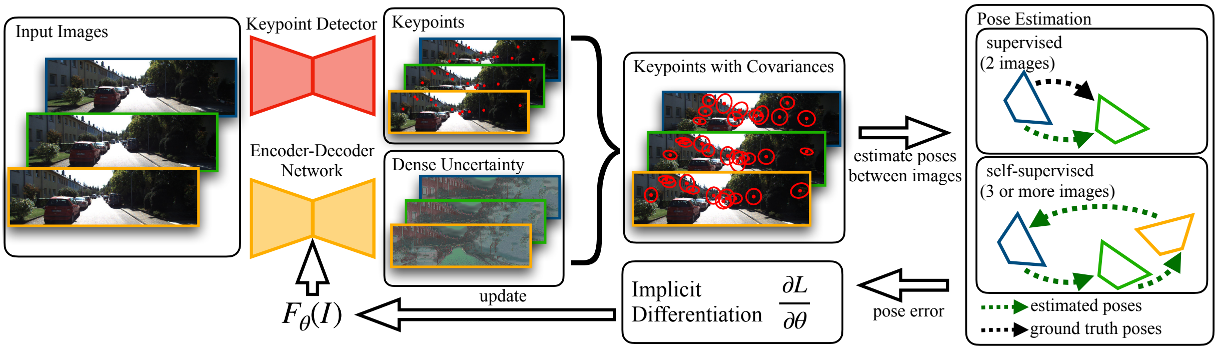


Figure 3. **Architecture**: We extract the uncertainty per image for every pixel using a UNet [55] backbone. Using keypoint locations from a keypoint detector, we obtain the keypoints with their estimated positional uncertainty. The relative pose is then estimated using a DNLS optimization. The UNet is updated by backpropagating the gradient (obtained by implicit differentiation) to the network output.

denoted by the superscript $^\top$. We deviate from [48] in the following: variables of the second frame are marked with the $'$ superscript, while variables of the first frame do not have a superscript. We represent the relative pose between images as a rigid-body transformation consisting of a rotation matrix $\boldsymbol{R} \in SO(3)$ and a unit length translation $\boldsymbol{t} \in \mathbb{R}^3$ ($\|\boldsymbol{t}\| = 1$ is imposed due to scale-invariance).

3.2. The Probabilistic Normal Epipolar Constraint

The asymetric probabilistic normal epipolar constraint (PNEC) estimates the relative pose, give two images $\boldsymbol{I}, \boldsymbol{I}'$ of the same scene under the assumption of uncertain feature positions in the second image. A feature correspondences is given by $\boldsymbol{p}_i, \boldsymbol{p}_i'$ in the image plane, where the uncertainty of $\boldsymbol{p}_i'$ is represented by the corresponding covariance $\boldsymbol{\Sigma}_{2\mathrm{D},i}'$. To get the epipolar geometry for the PNEC the feature points are unprojected using the camera intrinsics, giving unit length bearing vectors $\boldsymbol{f}_i, \boldsymbol{f}_i'$. The uncertainty of $\boldsymbol{f}_i'$ is now represented by $\boldsymbol{\Sigma}_i'$. Estimating the relative pose is done by minimizing the PNEC cost function as defined in [48]. For convenience we recap the energy function

$$E(\boldsymbol{R}, \boldsymbol{t}) = \sum_i \frac{e_i^2}{\sigma_i^2} = \sum_i \frac{|\boldsymbol{t}^\top (\boldsymbol{f}_i \times \boldsymbol{R}\boldsymbol{f}_i')|^2}{\boldsymbol{t}^\top \hat{\boldsymbol{f}}_i \boldsymbol{R} \boldsymbol{\Sigma}_i' \boldsymbol{R}^\top \hat{\boldsymbol{f}}_i{}^\top \boldsymbol{t}}, \quad (1)$$

in our notation. As mentioned previously, this asymmetric PNEC in [48] only considers uncertainties $\boldsymbol{\Sigma}'$ in the second frame. While this assumption might hold for the KLT tracking [66] used in [48], this leaves out important information when using other keypoint detectors like ORB [56] or SuperPoint [12]. Therefore, we will introduce a symmetric version of the PNEC that is more suitable for our task in the following.

Making the PNEC symmetric. As in [48] we assume the covariance of the bearing vectors $\boldsymbol{f}_i$ and $\boldsymbol{f}_i'$ to be gaussian, their covariance matrices denoted by $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_i'$, respectively. The new variance can be approximated as

$$\sigma_{s,i}^2 = \boldsymbol{t}^\top ((\widehat{\boldsymbol{R}\boldsymbol{f}_i'}) \boldsymbol{\Sigma}_i (\widehat{\boldsymbol{R}\boldsymbol{f}_i'})^\top + \hat{\boldsymbol{f}}_i \boldsymbol{R} \boldsymbol{\Sigma}_i' \boldsymbol{R}^\top \hat{\boldsymbol{f}}_i{}^\top) \boldsymbol{t}. \quad (2)$$

In the supplementary material (see App. C), we derive the variance and show the validity of this approximation given the geometry of the problem. This new variance now gives us the new symmetric PNEC with its following energy function

$$E_s(\boldsymbol{R}, \boldsymbol{t}) = \sum_i \frac{e_i^2}{\sigma_{s,i}^2} \quad (3)$$

3.3. DNLS for Learning Covariances

We want to estimate covariances $\boldsymbol{\Sigma}_{2\mathrm{D}}$ and $\boldsymbol{\Sigma}_{2\mathrm{D}}'$ (in the following collectively denoted as $\boldsymbol{\Sigma}_{2\mathrm{D}}$ for better readability) in the image plane

$$\boldsymbol{\Sigma}_{2\mathrm{D}} = \arg\min_{\boldsymbol{\Sigma}_{2\mathrm{D}}} \mathcal{L}, \quad (4)$$

such that they minimize a loss function $\mathcal{L}$ of the estimated pose. Since we found that the rotational error of the PNEC is more stable than the translational error, we chose to minimize only the rotational error

$$e_{\mathrm{rot}} = \angle \tilde{\boldsymbol{R}}^\top \boldsymbol{R} \quad (5)$$

$$\mathcal{L}(\tilde{\boldsymbol{R}}, \boldsymbol{R}; \boldsymbol{\Sigma}_{2\mathrm{D}}) = e_{\mathrm{rot}} \quad (6)$$

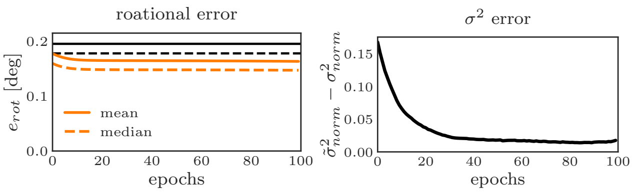
between the ground truth rotation $\tilde{\boldsymbol{R}}$ and the estimated rotation $\boldsymbol{R}$. We obtain

$$\boldsymbol{R} = \arg\min_{\boldsymbol{R}} E_s(\boldsymbol{R}, \boldsymbol{t}; \boldsymbol{\Sigma}_{2\mathrm{D}}) \quad (7)$$

by minimizing Eq. 3. To learn the covariances that minimize the rotational error, we can follow the gradient $d\mathcal{L}/d\boldsymbol{\Sigma}_{2\mathrm{D}}$. Implicit differentiation allows us to compute the gradient as

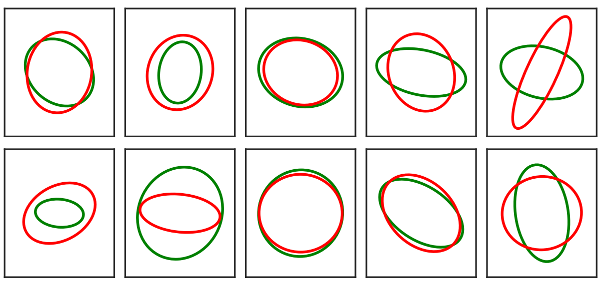
$$\frac{d\mathcal{L}}{d\boldsymbol{\Sigma}_{2\mathrm{D}}} = -\frac{\partial^2 E_s}{\partial \boldsymbol{\Sigma}_{2\mathrm{D}} \partial \boldsymbol{R}^\top} \left(\frac{\partial^2 E_s}{\partial \boldsymbol{R} \partial \boldsymbol{R}^\top} \right)^{-1} \frac{e_{\mathrm{rot}}}{\partial \boldsymbol{R}}. \quad (8)$$

For a detailed derivation of Eq. 8 and other methods, that unroll the optimization, to obtain the gradient we refer the interested reader to [13].

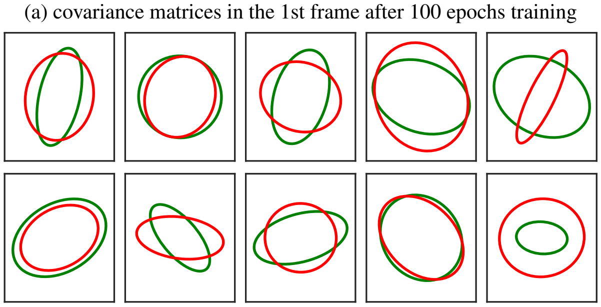


(a) e_{rot} during training (b) σ^2_{norm} error during training

Figure 4. Rotational error (a) and differences between the true residual variance $\tilde{\sigma}^2$ and the learned variance σ^2 (b) over the training epochs. Starting from uniform covariances, our method adapts the covariances for each keypoint to minimize the rotational error. Simultaneously, this leads to a better estimate of σ^2.



(a) covariance matrices in the 1st frame after 100 epochs training



(b) covariance matrices in the 2nd frame after 100 epochs training

Figure 5. Estimated (red) covariance ellipses in the first (a) and the second (b) frame, learned from 128 000 examples. Ground truth (green) covariances as comparison. Although the gradient minimizes the rotational error (see Fig. 4a), it is not capable of learning the correct covariance in the image plane.

Supervised Learning. The goal of the paper is for a neural network F learn the noise distributions of a keypoint detector. Given an image and a keypoint position, the network should predict the covariance of the noise $\mathbf{\Sigma}_{2\text{D},i} = F(\boldsymbol{I}, \boldsymbol{p}_i)$. The gradient $d\mathcal{L}/d\mathbf{\Sigma}_{2\text{D}}$ allows for the network to learn the covariance matrices in an end-to-end manner by regression on the relative pose error. Given a dataset with know ground truth poses, we can use

$$\mathcal{L}_{\text{sup}} = e_{\text{rot}} \tag{9}$$

as a training loss. This ensures, that learned covariances effectively minimize the rotational error. See Fig. 3 for overview of the training process.

Self-Supervised Learning. Finding a suitable annotated dataset for a specific task is often non-trivial. For our task, we need accurate ground truth poses that are difficult to acquire. But given a stream of images, like in VO, our method can be adapted to train a network in a self-supervised manner without the need for ground truth poses. For this, we follow the approach of [25] to exploit the cycle-consistency between a tuple of images. The cycle-consistency loss for a triplet $\{\boldsymbol{I}_1, \boldsymbol{I}_2, \boldsymbol{I}_3\}$ of images is given by

$$\mathcal{L}_{\text{cycl}} = \angle \prod_{(i,j)\in\mathcal{P}} \boldsymbol{R}_{ij} \,, \tag{10}$$

where $\boldsymbol{R}_{ij}$ is the estimated rotation between images I_i and I_j and $\mathcal{P} = \{(1,2),(2,3),(3,1)\}$ defines the cycle. As in [25], we also define an anchor loss

$$\mathcal{L}_{\text{anchor}} = \sum_{(i,j)\in\mathcal{P}} \angle \boldsymbol{R}_{ij}\boldsymbol{R}_{ij,\text{NEC}}^{\top} \tag{11}$$

with the NEC rotation estimate, as a regularising term. In contrast to [25], our method does not risk learning degenerate solutions from the cycle-consistency loss, since the rotation is estimated using independently detected keypoints. The final loss is then given by

$$\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{cycl}} + \lambda \mathcal{L}_{\text{anchor}} \,. \tag{12}$$

4. Experiments

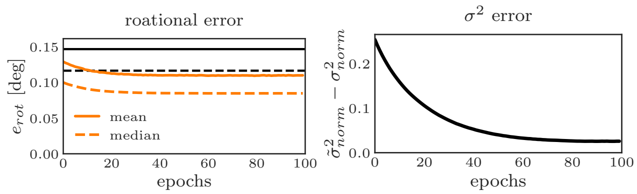
We evaluate our method in both synthetic and real-world experiments. Over the synthetic data, we investigate the ability of the gradient to learn the underlying noise distribution correctly by overfitting covariance estimates directly. We also investigate if better noise estimation leads to a reduces rotational error.

On real-world data, we use the gradient to train a network to predicts the noise distributions from images for different keypoint detectors. We explore fully supervised and self-supervised learning techniques for SuperPoint [12] and Basalt [66] KLT-Tracks to verify that our method is agnostic to the type of feature descriptor used (classical vs learned). We evaluate the performance of the learned covariances in a visual odometry setting on the popular KITTI odometry and the EuRoC dataset. We also evaluate generalization capabilities from the KITTI to the EuRoC dataset.

For our experiments we implement Eq. 3 in both Theseus [52] and ceres [1]. We use the Theseus implementation to train our network, since it allows for batched optimization and provides the needed gradient (see Eq. 8). However, we use the ceres implementation for our evaluation. We found the Levenberg-Marquardt optimization of ceres to be faster and more stable than its theseus counterpart.

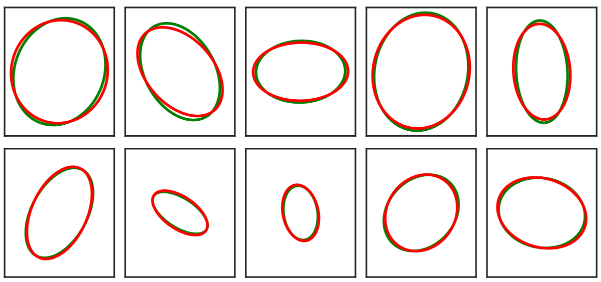
4.1. Simulated Experiments

In the simulated experiments we overfit covariance estimates for a single relative pose estimation problem using



(a) e_{rot} during training (b) σ^2_{norm} error during training

Figure 6. Rotational error (a) and differences between the true residual variance $\tilde{\sigma}^2$ and the learned variance σ^2 (b) over the training epochs. As previously, our method learns to adapt the covariances for each keypoint to minimize rotational error. Minimizing the rotational error leads to a significantly better estimate of σ^2.



(a) covariance matrices in the 2nd frame after 100 epochs training

Figure 7. Estimated (red) covariance ellipses in the second frame, learned from 128 000 examples. Ground truth (green) covariances as comparison. Training data with enough variety gives a gradient that allows to correctly learn the covariances even in the image plane, overcoming the unobservabilities of the first experiment.

the gradient from Eq. 8. For this, We create a random relative pose estimation problem consisting of two camera-frames observing randomly generated points in 3D space. The points are projected into camera frames using a pinhole camera model. Each projected point is assigned a random gaussian noise distribution. From this 128 000 random problems are sampled. We learn the noise distributions by initializing all covariance estimates as scaled identity matrices, solving the relative pose estimation problem using the PNEC and updating the parameters of the distribution using the gradient of Eq. 8 directly. We train for 100 epochs with the ADAM [31] optimizer with (0.9, 0.99) as parameters and a batch size of 12 800 for a stable gradient.

Fig. 4a shows the decrease of the rotation error over the epochs. The learned covariances decrease the error by 8% and 16% compared to unit covariances and the NEC, respectively. This validates the importance of good covariances for the PNEC, shown in [48]. Fig. 4b shows the average error for the normalized variance σ^2_{norm}, given by

$$\sigma^2_{i,norm} = \frac{N \cdot \sigma^2_i}{\sum_{j=0}^{N} \sigma^2_j} \tag{13}$$

over the training epochs, obtained at the ground truth relative pose. We compare the normalized error variance, as the scale of σ^2 is not observable from the gradient. The covariances that minimize the rotational error also approximate

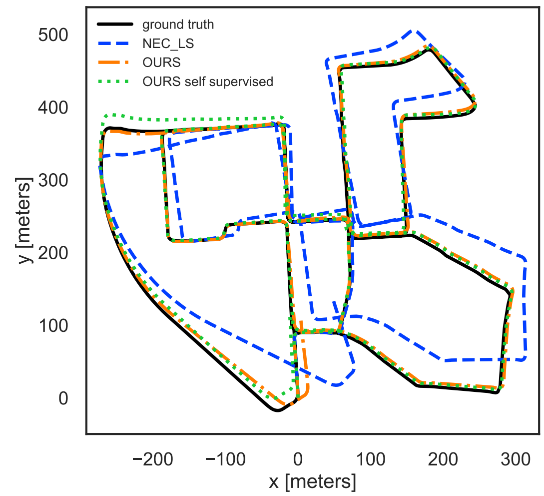


Figure 8. Qualitative trajectory comparison for KITTI seq. 00. Since we compare monocular methods, that cannot estimate the correct scale from a pair of images, we use the scale of the ground truth translations for visualization purposes. Both, our supervised and self-supervised approaches lead to significant improvements in the trajectory. There is little drift even without additional rotation averaging [11] or loop closure [49].

the residual uncertainty σ^2 very closely. However, while the residual uncertainty is approximated well, the learned 2D covariances in the image plane do not correspond to the correct covariances (see Fig. 5). This is due to two different reasons. First, due to σ_i^2 dependence on both $\mathbf{\Sigma}_{2\text{D},i}$ and $\mathbf{\Sigma}'_{2\text{D},i}$, there is not a single unique solution. Secondly, the direction of the gradient is dependent on the translation between the images (see App. D for more details). In this experimental setup, the information flow to the images is limited and we can only learn the true distribution for σ^2 but not for the 2D images covariances.

To address the problems with limited information flow of the previous experiment, we propose a second experiment to negate the influence of these aforementioned factors. First, each individual problem has a randomly sampled relative pose, where the first frame stays fixed. This removes the influence of the translation on the gradient direction. The noise is still drawn from the same distributions as earlier. Second, we fix the noise in the first frame to be small, isotropic, and homogeneous in nature. Furthermore, we only learn the covariances in the second frame and provide the optimization with the ground truth noise in the first frame. Fig. 6 and Fig. 7 show, that under these constraints, we are not only able the learn the distribution for σ^2 but also $\mathbf{\Sigma}'_{2\text{D}}$. Together, both experiments show, that we can learn the correct distributions from noisy data by following the gradient that minimizes the rotational error.

4.2. Real World Data

We evaluate our method on the KITTI [21] and EuRoC [9] dataset. Since KITTI shows outdoor driving sequences

	NISTÉR-5PT [50]			NEC [33]			NEC-LS			WEIGHTED NEC-LS			OURS SUPERVISED			OURS SELF-SUPERVISED		
Seq.	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t
08	0.195	17.020	4.24	0.081	8.284	3.66	0.056	7.004	2.50	0.054	6.059	2.50	**0.050**	**4.067**	2.46	0.050	4.118	**2.46**
09	0.142	5.754	1.74	0.053	1.646	1.43	0.052	1.553	0.71	0.051	1.354	**0.70**	0.049	1.317	0.71	**0.049**	**1.278**	0.70
10	0.295	16.678	6.57	0.167	9.264	4.43	0.064	4.787	1.79	**0.063**	4.389	1.76	0.063	**3.513**	**1.64**	0.065	3.821	1.65
train	0.249	11.506	4.13	0.141	10.127	2.97	0.082	6.910	1.72	0.081	6.410	1.72	0.077	**2.378**	**1.69**	**0.077**	2.505	1.69
test	0.200	14.349	4.07	0.089	6.917	3.28	0.056	5.353	1.96	0.055	4.676	1.95	**0.052**	**3.333**	1.91	0.053	3.408	**1.91**

Table 1. Quantitative comparison on the KITTI [21] dataset with SuperPoint [12] keypoints. We compare two rotation and one translation metric. The results are shown for each test sequence together with the mean results on the training and test set weighted by the sequence length. Both our training setups outperform the non-probablitic algorithms but also the weighted NEC-LS using SuperGlue confidences consistently across unseen data. The learned uncertainties are able to generalise well and improve the relative pose estimation significantly.

	NISTÉR-5PT [50]			NEC [33]			NEC-LS			KLT-PNEC [48]			OURS SUPERVISED			OURS SELF-SUPERVISED		
Seq.	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t
08	0.126	6.929	3.44	0.088	3.902	8.91	0.053	2.908	2.49	0.054	2.524	2.42	0.048	2.373	**2.36**	**0.047**	**1.706**	2.36
09	0.090	2.544	1.28	0.054	2.027	6.76	0.052	2.307	0.74	0.046	**1.003**	0.69	0.043	1.244	**0.64**	**0.042**	1.141	0.64
10	0.188	11.554	4.43	0.119	8.302	8.53	0.066	4.576	1.78	0.063	4.480	1.71	0.058	3.789	**1.58**	**0.056**	**3.623**	1.60
train	0.204	9.677	3.19	0.173	8.301	8.59	0.103	3.955	1.73	0.104	4.213	1.66	**0.094**	2.782	**1.60**	0.096	**2.737**	1.61
test	0.129	6.722	3.11	0.085	4.237	8.34	0.055	3.060	1.96	0.054	2.514	1.90	0.048	2.359	**1.82**	**0.048**	**1.910**	1.83

Table 2. Quantitative comparison on the KITTI [21] dataset with KLT tracks [66]. As in Tab. 1, we show the results on the test set together with the mean on the train and test set weighted by the sequence lengths. As for SuperPoint, our methods improve all metrics consistently for unseen data. Our learned covariances are significantly better for relative pose estimation than the approximation used in [48].

and EuRoC shows indoor scenes captured with a drone, they exhibit different motion models as well as a variety of images. For KITTI we choose sequences 00-07 as the training set for both supervised and self-supervised training. Sequences 08-10 are used as the test set. We use a smaller UNet [55] architecture as our network to predict the covariances for the whole image. We chose this network since it gives us a good balance between batch size, training time and performance. The network predicts the parameters for the covariances directly. We choose

$$\boldsymbol{\Sigma}_{2\mathrm{D}}(s,\alpha,\beta) = s\boldsymbol{R}_\alpha \begin{pmatrix} \beta & 0 \\ 0 & 1-\beta \end{pmatrix} \boldsymbol{R}_\alpha^\top \tag{14}$$

as a parameterization [7]. To ensure that our network predicts valid covariances the network output is filtered with

$$f_1(x) = (1+|x|)^{\mathrm{sign(x)}} \tag{15}$$
$$f_2(x) = x \tag{16}$$
$$f_3(x) = \frac{1}{1+e^{-x}} \tag{17}$$

for s, α, β, respectively. Feature points that have subpixel accuracy use the nearest pixel covariance. See App. E for more details on the training setup.

Supervised Learning. To show that our method generalizes to different keypoint detectors, we train two networks, one for SuperPoint [12] and one for KLT tracks obtained from [66]. The SuperPoint keypoints are matched using SuperGlue [57]. For training we use a batch size of 8 images pairs for SuperPoint and 16 images pairs for KLT tracks. We trained for 100 epochs for both SuperPoint and KLT tracks. More training details are provided in the supplementary material. To ensure our network does not overfit on specific keypoint locations, we randomly crop the images before finding correspondences during training time. During evaluation we use the uncropped images to obtain features. During training we randomly perturb the ground truth pose as a starting point. To increase robustness, we first use the eigenvalue based optimization of the NEC in a RANSAC scheme [32] to filter outliers. This is followed by a custom least squares implementation of the NEC (NEC-LS), followed by optimizing Eq. 3. As reported in [48] we found, that such a mutli-stage optimization provides the most robust and accurate results. We show examples of how the DNLS-learned covariances change the energy function landscape in the supplementary material.

Self-Supervised Learning. We evaluate our self-supervised training setup on the same data as our supervised training. Due to needing image tuples instead of pairs, we reduce the batch size to 12 for KLT image triplets. This gives us 24 and 36 images pairs per batch, respectively. The training epochs are reduced to 50. More training details for the supervised and self-supervised training can be found in the supplementary material.

Results. We evaluate the learned covariances in a VO setting. We compare the proposed DNLS approach to the

Seq.	Nistér-5pt [51] RPE$_1$	RPE$_n$	e_t	NEC [33] RPE$_1$	RPE$_n$	e_t	NEC-LS RPE$_1$	RPE$_n$	e_t	Weighted NEC-LS RPE$_1$	RPE$_n$	e_t	Ours self-supervised RPE$_1$	RPE$_n$	e_t	Ours Tab. 1 supervised RPE$_1$	RPE$_n$	e_t
V1_01	0.501	71.87	**31.86**	**0.320**	39.50	43.12	0.387	52.92	46.31	0.388	56.52	46.82	0.327	**31.12**	35.56	0.332	31.81	34.01
V1_02	0.541	32.01	**20.36**	**0.389**	**28.11**	26.95	0.540	70.08	28.94	0.542	68.35	29.81	0.444	30.39	21.98	0.436	29.07	21.29
V1_03	0.660	27.39	25.00	**0.492**	**25.42**	31.06	0.552	76.72	31.58	0.555	78.14	32.25	0.510	29.52	24.19	0.520	31.18	**24.13**
V2_01	0.515	61.45	33.51	0.316	31.95	39.79	0.310	35.84	39.00	0.314	38.62	39.62	**0.285**	**17.61**	32.40	0.295	22.41	**30.58**
V2_02	0.545	43.73	22.24	0.396	25.48	32.21	0.369	26.96	25.36	**0.365**	25.09	25.81	0.382	25.32	21.16	0.386	**21.91**	**20.34**
V2_03	1.123	**36.71**	**28.77**	0.976	48.26	37.60	**0.939**	107.11	36.74	0.941	100.73	36.71	0.942	52.72	31.13	0.991	55.41	30.40
mean	0.631	48.45	27.56	0.463	33.51	36.03	0.494	58.90	35.61	0.496	58.95	36.11	**0.461**	**30.57**	28.46	0.472	31.44	**27.44**

Table 3. Quantitative comparison on the Vicon sequences of the EuRoC dataset [9] with SuperPoint [12] keypoints. The dataset is more difficult than KITTI (see Tab. 2 and Tab. 1) with SuperPoint and SuperGlue [57] finding far fewer matches. As reported in [48] the least squares implementations struggle with bad initialization under these adverse conditions with NEC-LS performing especially poor. From all least squares optimizations, our learned covariances consistently perform the best, even outperforming the NEC most of the time.

popular 5pt algorithm [51] and the NEC [33] as implemented in [32]. To investigate the benefit of our learned covariances we include the NEC-LS implementation as well as the symmetric PNEC with the covariances from [48] in Tab. 2. For Tab. 1 we additionally include a weighted version of our custom NEC-LS implementation with matching confidence from SuperGlue as weights. All methods are given the same feature matches and use a constant motion model for initializing the optimizations. We evaluate on the rotational versions of the RPE$_1$ and RPE$_n$ and the cosine error e_t for the translation as defined in [11, 48]. Tab. 1 and Tab. 2 show the average results on the test set over 5 runs for SuperPoint and KLT tracks on KITTI [21], respectively. We show additional results in App. G. Our methods consistently perform the best over all sequences, with the self-supervised being on par with our supervised training. Compared to its non-probabilistic counterpart NEC-LS, our method improves the RPE$_1$ by 7% and 13% and the RPE$_n$ by 37% and 23% for different keypoint detectors on unseen data. It also improves upon weighted methods, like weighted NEC-LS and the non-learned covariances for the PNEC [48]. This demonstrates the importance of correctly modeling the feature correspondence quality. We show an example trajectory in Fig. 8.

Tab. 3 shows the results on the EuRoC dataset for SuperPoint. Pose estimation is significantly more difficult compared to KITTI, often having few correspondences between images. However, our method generalizes to different datasets, with the network trained on KITTI and our self-supervised approach, outperforming the others most of the time. Especially a direct comparison with NEC-LS, the closest non-probabilistic method, shows significant improvements of 7% for RPE$_1$ and 48% for the RPE$_n$.

5. Discussion and Limitations

Our experiments demonstrate the capability of our framework to to correctly learn positional uncertainty, leading to improved results for relative pose estimation for VO. Our approach generalizes to different feature extractors and to different datasets, providing a unified approach to estimate the noise distribution of keypoint detectors. However, our method requires more computational resources than the original uncertainty estimation for the PNEC.

We evaluate our learned covariances in a visual odometry setting, showing that they lead to reduced errors and especially less drift in the trajectory. However, this does not guarantee that the covariances are *calibrated*. Our framework inherits the ambiguity of the PNEC with regard to the noise scale. The true scale of the noise is not observable from relative pose estimation alone and only the relative scale between covariances can be learned. For the purposes of VO, this scale ambiguity is negligible.

As our synthetic experiments show, diverse data is needed to correctly identify the 2D noise distribution. However, obtaining the noise distribution is difficult for keypoint detectors – hence learning it from pose regression. Further limitations are addressed in App. B.

6. Conclusion

We present a novel DNLS framework for estimating positional uncertainty. Our framework can be combined with any feature extraction algorithm, making it extremely versatile. Regressing the noise distribution from relative pose estimation, ensures that learned covariance matrices are suitable for visual odometry tasks. In synthetic experiments, our framework is capable to learn the correct noise distribution from noisy data. We showed the practical application of our framework on real-world data for different feature extractors. Our learned uncertainty consistently outperforms a variety of non-probabilistic relative pose estimation algorithms as well as other uncertainty estimation methods.

Acknowledgements. This work was supported by the ERC Advanced Grant SIMULACRON, by the Munich Center for Machine Learning and by the EPSRC Programme Grant VisualAI EP/T028572/1.

Learning Correspondence Uncertainty via Differentiable Nonlinear Least Squares

Supplementary Material

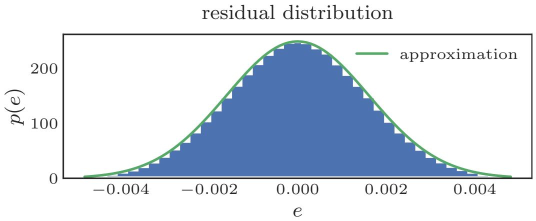


Figure 9. Approximation of the residual variances. The analytical approximation given in the main paper accurately models the true distribution of the residual given a similar setup to the KITTI dataset.

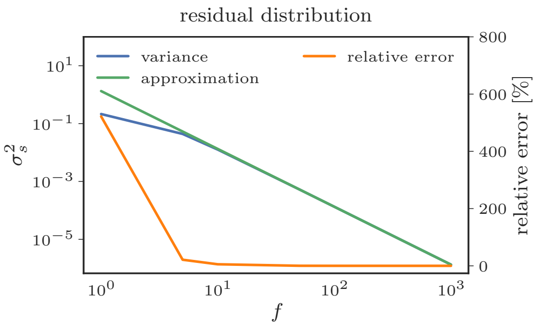


Figure 10. Approximation of the residual variances over different focal lengths. The scale of the variance is correlated with the focal length. Our approximation is better, the smaller the variance is. For a focal length similar to the one found in the KITTI dataset, the relative error is 0.015%.

A. Overview

This supplementary material presents additional insight into learning positional uncertainty using DNLS. We start by addressing limitations of our framework in App. B. App. C gives a derivation of the residual variance σ_s^2 for the symmetric PNEC. We investigate the unobservabilities of the gradient in App. D. The training and evaluation details are given in App. E. We show further quantitative evaluations in App. F and App. G. This includes examples of how the learned covariances move the minimum around the ground truth and the results on the sequences 00-07 of the KITTI [21] dataset. We compare our learned covariances against error estimates from reprojection using ground truth poses.

B. Limitations

In this section, we will address limitations of our method, not mentioned in the main paper due to constrained space. We learn to estimate the noise distribution of keypoint detectors, using regression on the pose error. The gradient we use for learning the distribution is restricted to points in the image that are detected as keypoints. This restrict our method to learn only on regions of the image with a high chance of producing keypoints. While we don't need uncertainty information for regions without keypoints, this sparse information flow might reduce generalization capabilities to different datasets. Sparsity if further enhanced by using RANSAC to filter outliers, removing points that are too far off. However, we choose to include RANSAC for our training to obtain better pose estimates for gradients not dominated by outliers. We tried to mitigate the effect of overfitting on keypoint positions by cropping the images, leading to different keypoint positions. Furthermore, our experiments showed that generalization between KITTI and EuRoC are possible.

Fig. 11 and Fig. 12 show examples where our method performs worse and better than the NEC-LS optimization based on the estimated covariances. We investigate the keypoints with the highest and lowest reprojection error. As Fig. 11 shows, our method is not always able to compensate keypoints on dynamic objects leading to a large rotational error. The trajectories in Fig. 12 show the improvements our method is able to achieve compared to NEC-LS.

C. Approximating σ_s^2

This section show derives the residual variance from the bearing vector covariances in both images. Given both bearing vectors $\boldsymbol{f}$ and $\boldsymbol{f\prime}$ are noisy, we can write them as

$$\boldsymbol{f} = \boldsymbol{\mu} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})\,, \tag{18}$$

$$\boldsymbol{f} = \boldsymbol{\mu}' + \boldsymbol{\eta}', \quad \boldsymbol{\eta}' \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}')\,, \tag{19}$$

with a constant and a noise term. We then get the new normal vector as

$$\begin{aligned} \boldsymbol{n}_s &= (\boldsymbol{\mu} + \boldsymbol{\eta}) \times \boldsymbol{R}(\boldsymbol{\mu}' + \boldsymbol{\eta}') \\ &= \hat{\boldsymbol{\mu}}\boldsymbol{R}\boldsymbol{\mu}' + \hat{\boldsymbol{\mu}}\boldsymbol{R}\boldsymbol{\eta}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\mu}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\eta}'\,, \end{aligned} \tag{20}$$

with a constant term $\boldsymbol{\mu}_n = \hat{\boldsymbol{\mu}}\boldsymbol{R}\boldsymbol{\mu}'$ and a noise term $\boldsymbol{\eta}_n = \boldsymbol{R}\boldsymbol{\eta}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\mu}' + \hat{\boldsymbol{\eta}}\boldsymbol{R}\boldsymbol{\eta}'$. The noise term is zero centered and has a variance of

$$\boldsymbol{\Sigma}_n = (\widehat{\boldsymbol{R}\boldsymbol{\mu}'_i})\boldsymbol{\Sigma}_i(\widehat{\boldsymbol{R}\boldsymbol{\mu}'_i})^\top + \hat{\boldsymbol{\mu}}_i\boldsymbol{R}\boldsymbol{\Sigma}'_i\boldsymbol{R}^\top\hat{\boldsymbol{\mu}}_i^\top + \tilde{\boldsymbol{\Sigma}}\,, \tag{21}$$

where $\tilde{\boldsymbol{\Sigma}}$ is constructed from the columns of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}'_R = \boldsymbol{R}\boldsymbol{\Sigma}'\boldsymbol{R}^\top$ as

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} (\boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}'_{R,3} + \boldsymbol{\Sigma}_3 \times \boldsymbol{\Sigma}'_{R,2})^\top \\ (\boldsymbol{\Sigma}_3 \times \boldsymbol{\Sigma}'_{R,1} + \boldsymbol{\Sigma}_1 \times \boldsymbol{\Sigma}'_{R,3})^\top \\ (\boldsymbol{\Sigma}_1 \times \boldsymbol{\Sigma}'_{R,2} + \boldsymbol{\Sigma}_2 \times \boldsymbol{\Sigma}'_{R,1})^\top \end{bmatrix}\,. \tag{22}$$

As stated in the main paper, we use an approximation of the noise distribution. Since $\tilde{\boldsymbol{\Sigma}}$ is order of magnitudes smaller than the other terms, we can approximate $\boldsymbol{\Sigma}_n$ as

$$\boldsymbol{\Sigma}_n \approx (\widehat{\boldsymbol{R}\boldsymbol{\mu}'_i})\boldsymbol{\Sigma}_i(\widehat{\boldsymbol{R}\boldsymbol{\mu}'_i})^\top + \hat{\boldsymbol{\mu}}_i\boldsymbol{R}\boldsymbol{\Sigma}'_i\boldsymbol{R}^\top\hat{\boldsymbol{\mu}}_i^\top\,. \tag{23}$$

The final residual variance is given by

$$\sigma_s^2 = \boldsymbol{t}^\top\boldsymbol{\Sigma}_n\boldsymbol{t}\,. \tag{24}$$

Fig. 9 shows a comparison between our approximation and a the true residual distribution, given noisy image points. Do to the unprojection of the image points to bearing vectors, the trace of the bearing vector covariances is small for a focal length f of ca. 720 pixels on the KITTI dataset, since $\mathrm{tr}(\boldsymbol{\Sigma}) \sim 1/f^2$. Given the small covariances, $\tilde{\boldsymbol{\Sigma}}$ is several magnitudes smaller than the other terms, making the approximation accurate. Fig. 10 shows the correlation between the variance and the focal length.

D. Gradient

In this section, we show that the gradient $\partial\mathcal{L}/\partial\boldsymbol{\Sigma}_{\mathrm{2D}}$ is restricted by the problem geometry. We state the components needed to obtain $\partial\mathcal{L}/\partial\boldsymbol{\Sigma}_{\mathrm{2D}}$ and show, how the geometry restricts their direction. Therefore, given a constant geometry the overall gradient direction only moves little throughout the training.

We start by rewriting the residual e_s of symmetric PNEC energy function as

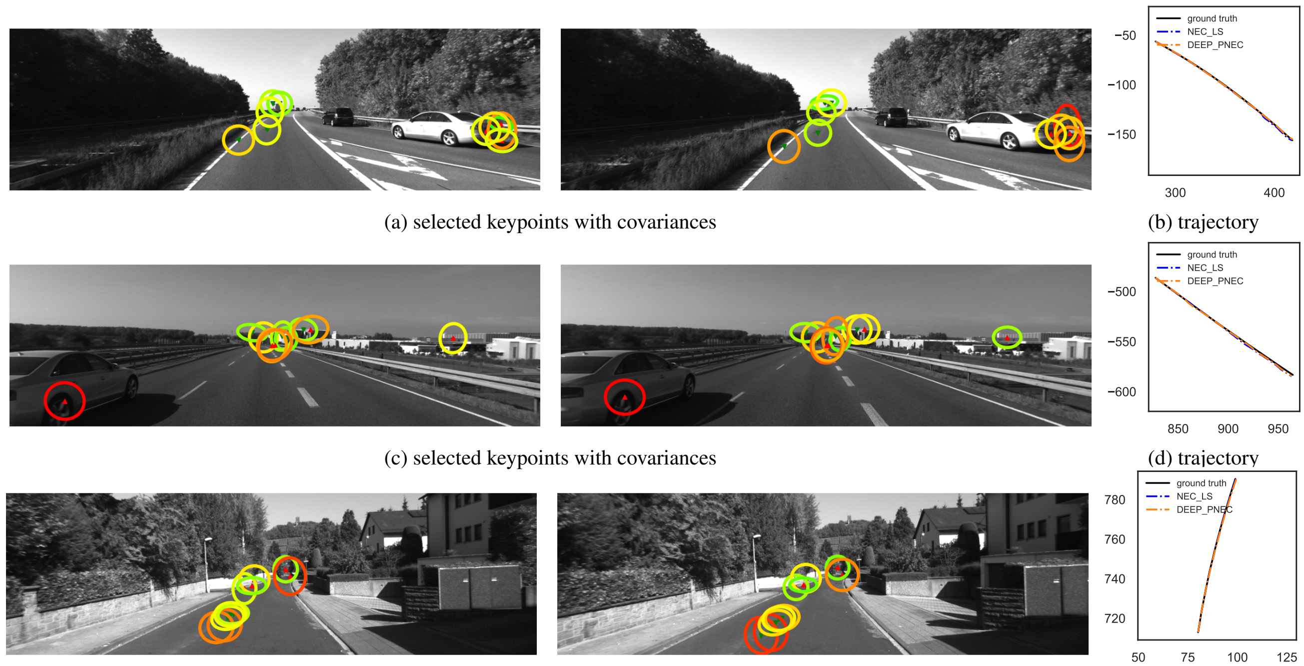
$$e_s = \frac{n}{\sigma_s} = \frac{n}{\sqrt{d_{\boldsymbol{\Sigma}} + d_{\boldsymbol{\Sigma}'}}}\,, \tag{25}$$

for easier differentiation, with the components

$$n = \boldsymbol{t}^\top\hat{\boldsymbol{f}}\exp\hat{\boldsymbol{x}}\boldsymbol{R}\boldsymbol{f}'\boldsymbol{f}'^\top\boldsymbol{R}^\top\exp\hat{\boldsymbol{x}}^\top\hat{\boldsymbol{f}}^\top\boldsymbol{t}\,, \tag{26}$$

$$d_{\boldsymbol{\Sigma}} = \left(\left(\exp\hat{\boldsymbol{x}}\boldsymbol{R}\boldsymbol{f}'\right) \times \boldsymbol{t}\right)^\top\boldsymbol{\Sigma}\left(\left(\exp\hat{\boldsymbol{x}}\boldsymbol{R}\boldsymbol{f}'\right) \times \boldsymbol{t}\right)\,, \tag{27}$$

$$d_{\boldsymbol{\Sigma}'} = \boldsymbol{t}^\top\hat{\boldsymbol{f}}\exp\hat{\boldsymbol{x}}\boldsymbol{R}\boldsymbol{\Sigma}'\boldsymbol{R}^\top\exp\hat{\boldsymbol{x}}^\top\hat{\boldsymbol{f}}^\top\boldsymbol{t}\,. \tag{28}$$



(a) selected keypoints with covariances (b) trajectory

(c) selected keypoints with covariances (d) trajectory

(e) selected keypoints with covariances (f) trajectory

Figure 11. Left: estimated keypoints with covariances (color-coded ellipses) for examples where our method performs worse than NEC-LS. Good (▼) and bad correspondences (▲) based on the reprojection error. Right: corresponding sections of the trajectory. (a) and (c) show examples with keypoints on dynamic objects. Although their estimated covariances is somewhat lower (especially in (c)) this is not enough to compensate the error. (e) shows an example where points with a higher reprojection error get assigned a covariances on a similar level or slightly better than good correspondences.

Since we are working with rotations in $\mathrm{SO}(3)$ we differentiate with regard to $\boldsymbol{x} \in \mathrm{so}(3)$ around the identity rotation. This gives us the following gradients

$$\frac{\partial n}{\partial \boldsymbol{x}} = 2\left((\boldsymbol{R}\boldsymbol{f}'\boldsymbol{f}'^\top \boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t}) \times (\hat{\boldsymbol{f}}\boldsymbol{t}) \right)^\top , \tag{29}$$

$$\frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{x}} = 2\left((\boldsymbol{R}\boldsymbol{f}') \times (\hat{\boldsymbol{t}}\boldsymbol{\Sigma}\hat{\boldsymbol{t}}^\top \exp \hat{\boldsymbol{x}}\boldsymbol{R}\boldsymbol{f}') \right)^\top , \tag{30}$$

$$\frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{x}} = 2\left((\boldsymbol{R}\boldsymbol{\Sigma}'\boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t}) \times (\hat{\boldsymbol{f}}\boldsymbol{t}) \right)^\top , \tag{31}$$

with regard to the rotation. The direction of each gradient is restricted by the cross product. The gradient for the residual is given by

$$\frac{\partial e_s}{\partial \boldsymbol{x}} = \frac{1}{\sigma_s}\frac{\partial n}{\partial \boldsymbol{x}} - \frac{n}{2\sigma_s^3}\left(\frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{x}} + \frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{x}} \right) . \tag{32}$$

The gradients with regard to the bearing vector covariances are solely dependent on the geometry as they are given by

$$\frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{\Sigma}} = \left(\boldsymbol{t} \times (\exp \hat{\boldsymbol{x}}\boldsymbol{R}\boldsymbol{f}')\right)\left(\boldsymbol{t} \times (\exp \hat{\boldsymbol{x}}\boldsymbol{R}\boldsymbol{f}')\right)^\top , \tag{33}$$

$$\frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{\Sigma}'} = \left(\boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t}\right)\left(\boldsymbol{R}^\top \exp \hat{\boldsymbol{x}}^\top \hat{\boldsymbol{f}}^\top \boldsymbol{t}\right)^\top . \tag{34}$$

The gradients of the residual are given by

$$\frac{\partial e_s}{\partial \boldsymbol{\Sigma}} = -\frac{n}{2\sigma_s^3}\frac{\partial d_{\boldsymbol{\Sigma}}}{\partial \boldsymbol{\Sigma}} , \tag{35}$$

$$\frac{\partial e_s}{\partial \boldsymbol{\Sigma}'} = -\frac{n}{2\sigma_s^3}\frac{\partial d_{\boldsymbol{\Sigma}'}}{\partial \boldsymbol{\Sigma}'} . \tag{36}$$

Since all components are restricted by the geometry of the problem, the overall gradient is somewhat restricted as well. We show this empirically in the following.

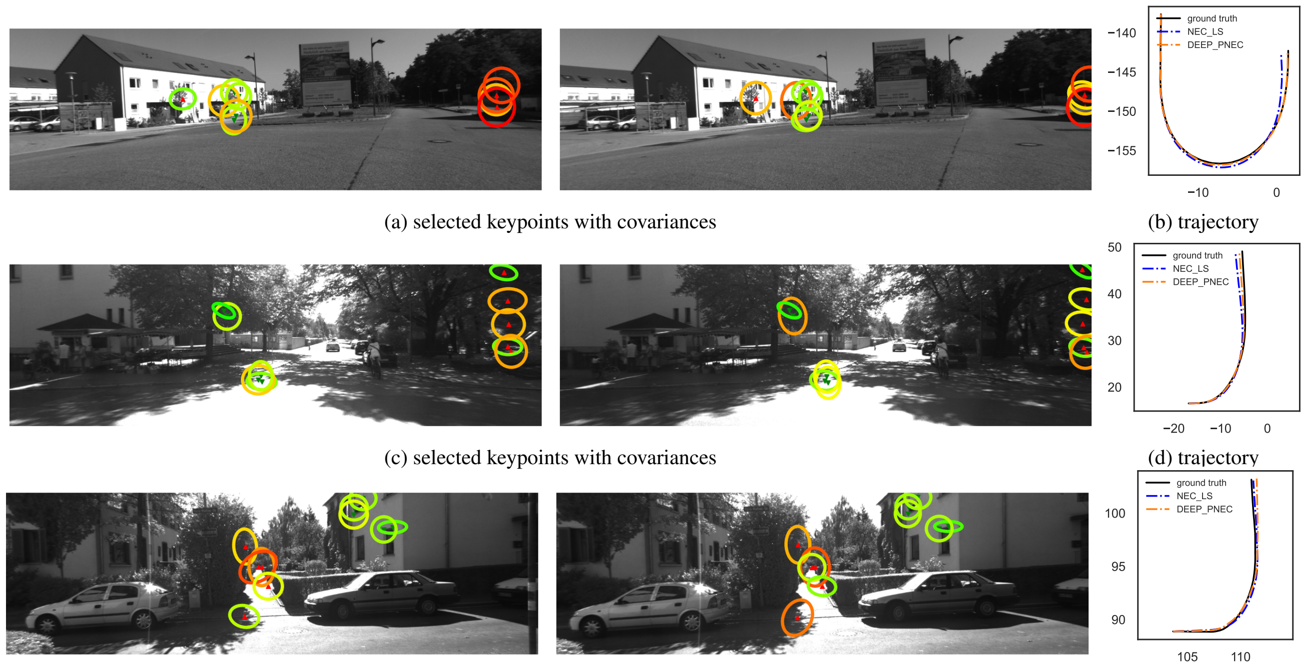
Fig. 13 and Fig. 14 give the distribution of the gradient for the first experiment on synthetic data, where all individual problems share the same geometric setup. Fig. 14 shows the eigenvectors of $\partial\mathcal{L}/\partial\boldsymbol{\Sigma}_{2\mathrm{D}}$ for one covariance in the image plane. After 10 epochs of training, the eigenvectors are mainly located at 4 distinct regions, showing the restriction of the gradient direction. Even after 100 epochs of training certain regions show only few eigenvectors. The angular distribution of the eigenvectors in Fig. 13 show 4 distinct peaks, with almost no eigenvectors in between.

Fig. 15 and Fig. 16 show the distribution of the gradient for the second experiment on synthetic data, with more diverse data. Given the diverse data, there are eigenvectors in all directions, even after 10 epochs. Fig. 15 still shows 4 distinct peaks, however there is no sparsity in the distribution.

The sparse distribution of the gradient direction prohibit learning the correct noise distribution for the first experiment. Only the residual variance is correctly estimated. However, the introduction of diverse data with different geometries removes this restriction, leading better covariance estimates.

E. Hyperparameters

This section details the training and evaluation parameters for our DNLS framework for estimating noise distributions of keypoints. All models are trained on two RTX 5000 GPUs with 16GB of memory for around 3 days. We use a UNet architecture with 3 output channels for predicting the uncertainty parameters. The UNet has 4 down convolutions and 4 up convolutions with 32, 64, 128, 256 and 128, 64, 32, 16 channels, respectively. Tab. 5 gives the SuperPoint and SuperGlue hyperparameters for training and evaluation. For our supervised training, we train on consecutive image pairs of the training sequences. For our self-supervised training we create the training tuples from 3 consecutive images. When training with SuperPoint, we crop the images to size (1200, 300), whereas



(a) selected keypoints with covariances (b) trajectory

(c) selected keypoints with covariances (d) trajectory

(e) selected keypoints with covariances (f) trajectory

Figure 12. Left: estimated keypoints with covariances (color-coded ellipses) for examples where our method performs better than NEC-LS. Good (▼) and bad correspondences (▲) based on the reprojection error. Right: corresponding sections of the trajectory. Covariances for bad correspondences are estimated to be higher in these examples. They are down-weighted in the optimization leading to better pose estimates.

Hyperparameter	KITTI	EuRoC
optimizer	ADAM	ADAM
β_1	0.9	0.9
β_2	0.999	0.999
learning rate	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
PNEC and theseus		
regularization	10^{-13}	10^{-13}
damping	10^{7}	10^{7}
iterations	100	100
RANSAC		
iterations	5000	5000
threshold	10^{-6}	$8 \cdot 10^{-7}$

Table 4. Parameters used for training and evaluation.

for KLT-Tracks, we crop it to (1200, 320). We found that reducing the height too much for KLT-tracks leads to not enough tracks. For evaluating with KLT-tracks on KITTI we change the following to [48]: instead of tracking keypoints over multiple images, we start with fresh keypoints for each image pair. To account for the symmetric PNEC, we slightly modify the uncertainty extraction. We use [48, suppl., Eqn. (8)] as the uncertainty measure for the tracks in both frames. We found, that these changes already give better results than the ones stated [48]. Tab. 4 gives the training parameter for optimizer, theseus and the PNEC energy function not stated in the main paper.

Hyperparameter	training	KITTI	EuRoC
max keypoints	256	2048	1024
keypoint threshold	0.005	0.005	0.0005
nms radius	3	3	3
weights	outdoor	outdoor	indoor
sinkhorn iterations	20	20	20
match threshold	0.5	0.5	0.01

Table 5. Hyperparameters for SuperPoint and SuperGlue during training and evaluation on the KITTI and EuRoC dataset.

F. Moving the Minimum

Fig. 18 and Fig. 17 show examples for energy functions around the ground truth pose on the KITTI dataset. The energy functions are evaluated with keypoints filtered using the reprojection error also used in the RANSAC scheme of [32] to remove outliers. We show the energy functions evaluated for rotations around the ground truth for yaw and pitch. While the overall shape of the energy function stays the same, our methods moves the minimum closer to the ground truth pose by learning the covariances.

G. Further Results

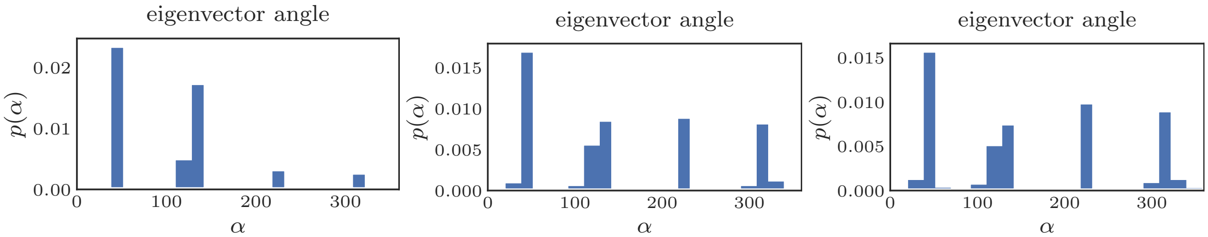
In this section we present additional results on the KITTI dataset, not presented in the main paper due to constrained space. We give the evaluation results for all sequences, training and test set. To present more comparisons with baseline methods, we replace the Nistér-5pt [51] with the 8pt [42] algorithm. Furthermore, we replace the weighted NEC-LS and the KLT-PNEC. Instead, we add another PNEC method, where we approximate the error distribution using a reprojection error. Following [32], we triangulate a 3D point using the feature correspondence $\boldsymbol{p}_i, \boldsymbol{p}'_i$ and the ground truth pose. We reproject the point into the images as $\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}'_i$ and approximate the the error distribution as scaled isotropic covariances

$$\boldsymbol{\Sigma}_{2\mathrm{D},i} = \|\tilde{\boldsymbol{p}}_i - \boldsymbol{p}_i\|^2 \boldsymbol{I}_2 \,, \tag{37}$$

$$\boldsymbol{\Sigma}'_{2\mathrm{D},i} = \|\tilde{\boldsymbol{p}}'_i - \boldsymbol{p}'_i\|^2 \boldsymbol{I}_2 \,. \tag{38}$$

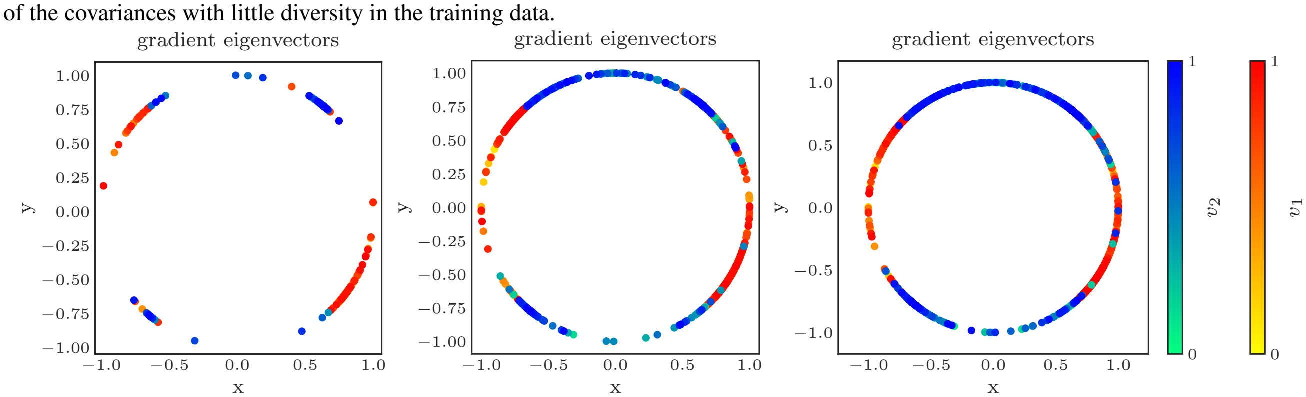
We clip the scale of the covariances at 0.01 and 4.0. Tab. 7 shows the results for the training and test set on KITTI with SuperPoint. While the reprojection method achieves the best results for the RPE_1 and e_t, our methods are often not far behind. This shows, that our network is capable and not too far off, when it comes to pose estimation. Tab. 6 shows the results for KITTI with KLT-tracks.

We show trajectories for all sequences of the KITTI dataset in Fig. 20 and Fig. 19. Our method consistently achieves the smallest drift over all sequences.



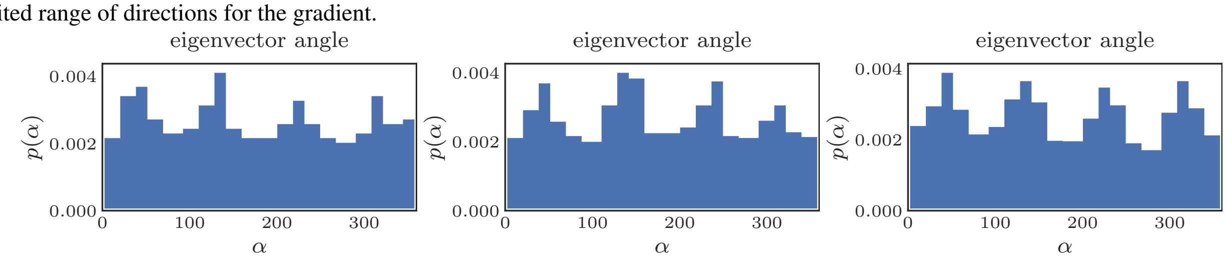
(a) 10 epochs (b) 50 epochs (c) 100 epochs

Figure 13. Histogram of eigenvector angles for the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2\mathrm{D}}$ after 10, 50, and 100 epochs. The histogram shows 4 distinct peaks, with only a few points in between. This shows the limited direction that the gradients have, making it difficult to learn the true distribution of the covariances with little diversity in the training data.



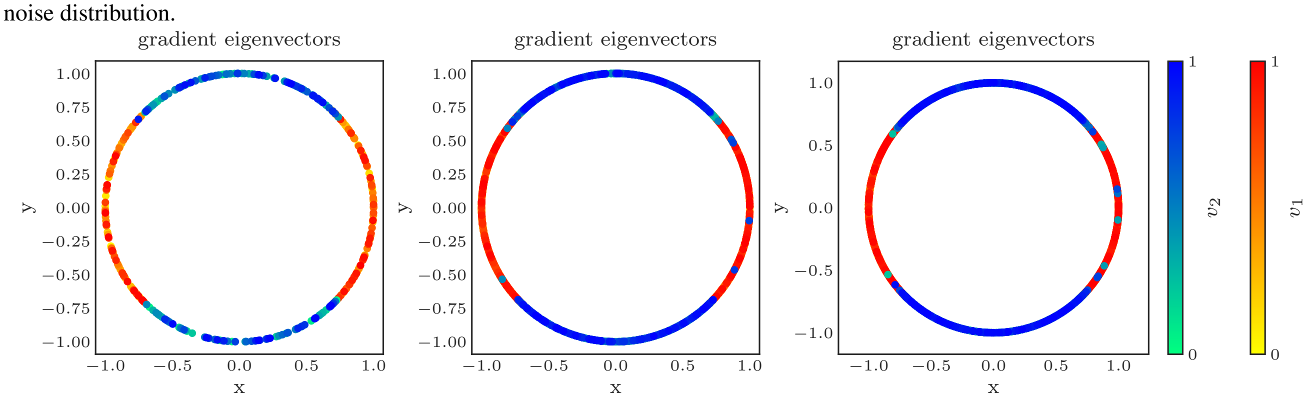
(a) 10 epochs (b) 50 epochs (c) 100 epochs

Figure 14. Distribution of eigenvectors of the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2\mathrm{D}}$ after 10, 50, and 100 epochs. Eigenvectors are color coded (green to blue and yellow to red) depending, whether there are the 1st or 2nd eigenvector and their epoch. While after 100 epochs most of the circle is covered, the eigenvectors aggregate at certain positions. Especially after 10 epochs, the eigenvectors are sparsely distributed. This shows a limited range of directions for the gradient.



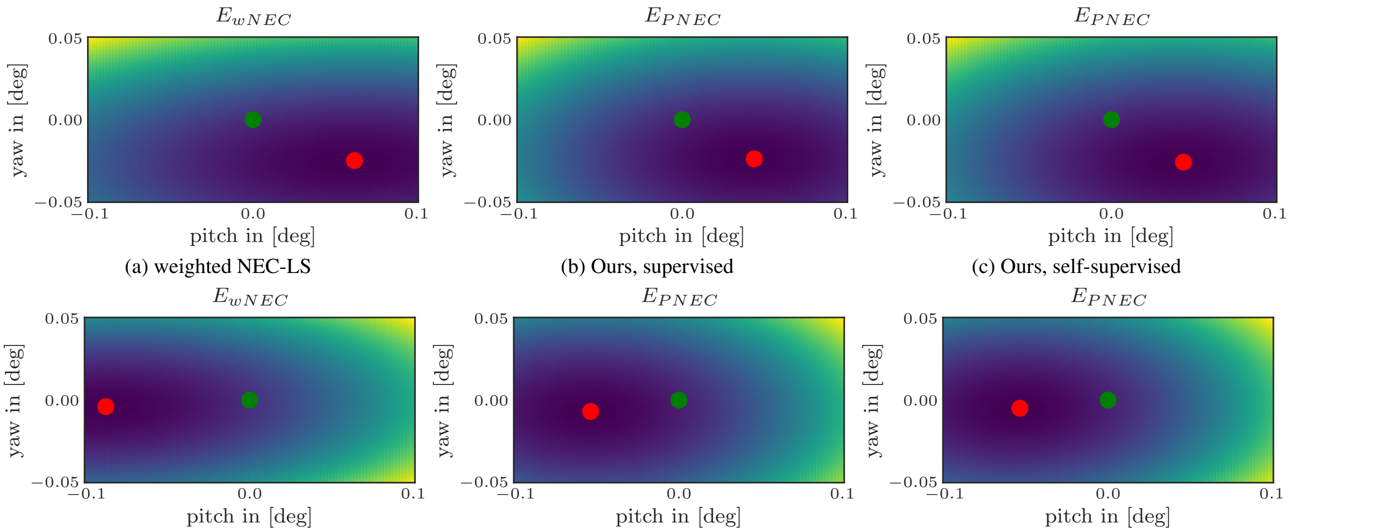
(a) 10 epochs (b) 50 epochs (c) 100 epochs

Figure 15. Histogram of eigenvector angles for the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2\mathrm{D}}$ after 10, 50, and 100 epochs. While it shows 4 distinct peaks, event after only 10 epochs many points lie in between. The direction of the gradient is not limited, allowing for a better fit to the ground truth noise distribution.



(a) 10 epochs (b) 50 epochs (c) 100 epochs

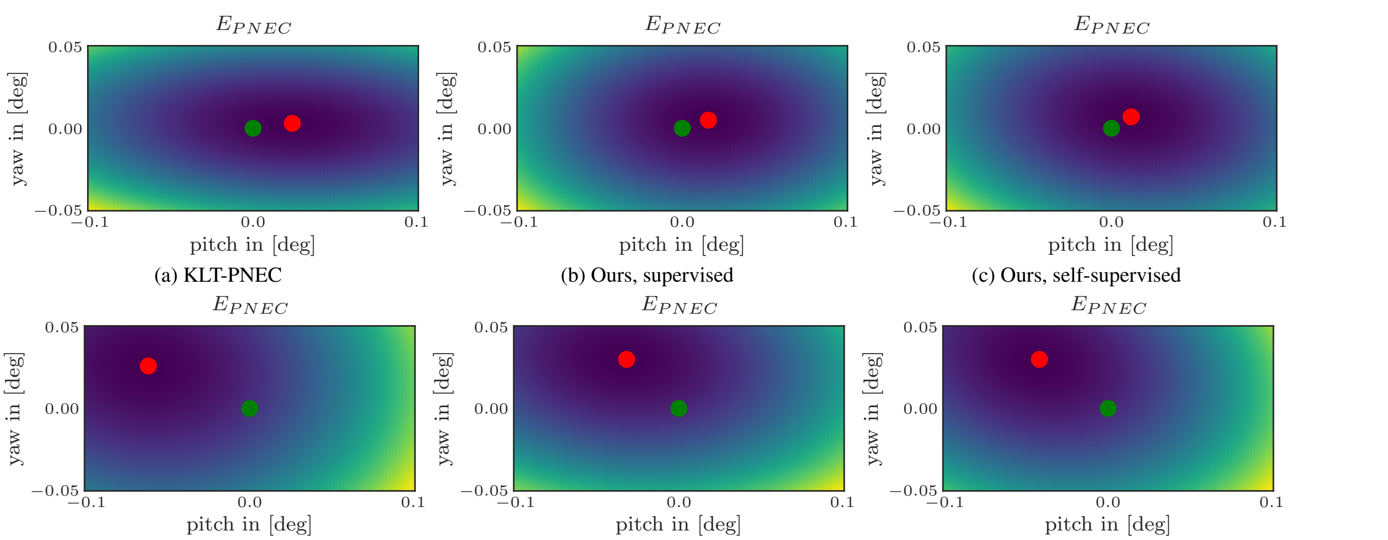
Figure 16. Distribution of eigenvectors of the gradient $\partial\mathcal{L}/\partial\mathbf{\Sigma}'_{2\mathrm{D}}$ after 10, 50, and 100 epochs. Eigenvectors are color coded (green to blue and yellow to red) depending, whether there are the 1st or 2nd eigenvector and their epoch. Even after 10 epochs, the eigenvectors are evenly distributed. This show, that the gradient has no limit for its direction, allowing for a better fit to the noise distribution even in the image plane.



(a) weighted NEC-LS (b) Ours, supervised (c) Ours, self-supervised

(d) weighted NEC-LS (e) Ours, supervised (f) Ours, self-supervised

Figure 17. Energy functions evaluated for rotations around the ground truth pose (green). Minimum of the cost function is marked in red. The energy function is evaluated for SuperPoint keypoint for two pose estimation problems on the KITTI dataset, filtered with RANSAC at the ground truth pose. We compare the weighted NEC-LS energy function to the PNEC energy function with our supervised and self-supervised covariances. While the overall shape of the energy function stays the same, our learned covariances move the minimum closer to the ground truth.



(a) KLT-PNEC (b) Ours, supervised (c) Ours, self-supervised

(d) KLT-PNEC (e) Ours, supervised (f) Ours, self-supervised

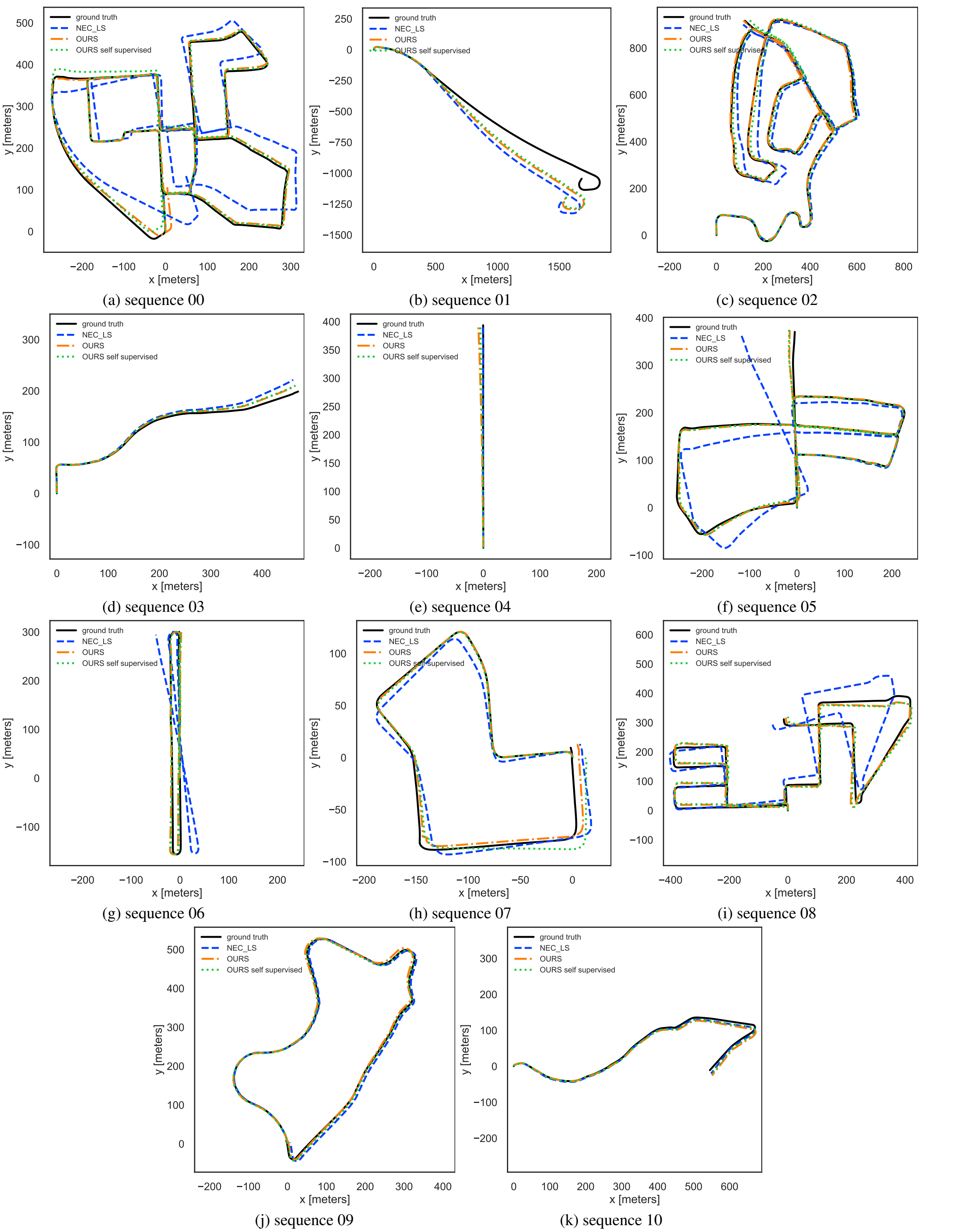
Figure 18. Energy functions evaluated for rotations around the ground truth pose (green). Minimum of the cost function is marked in red. The energy function is evaluated for KLT-tracks for two pose estimation problems on the KITTI dataset, filtered with RANSAC at the ground truth pose. We compare the PNEC energy function using the KLT-covariances with our supervised and self-supervised covariances. While the overall shape of the energy function stays the same, our learned covariances move the minimum closer to the ground truth.

	8PT [42]			NEC [33]			NEC-LS			OURS SUPERVISED			OURS SELF-SUPERVISED			REPROJECTION		
Seq.	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t
00	0.185	7.203	2.61	0.153	5.505	9.32	0.121	**2.403**	1.42	0.115	2.994	1.31	**0.113**	3.110	**1.30**	0.117	3.080	1.29
01	**0.253**	7.162	2.89	0.659	28.523	5.24	0.270	8.991	**2.20**	0.294	6.433	2.23	0.349	**6.042**	2.27	0.363	7.712	2.20
02	0.159	7.451	1.85	0.115	6.891	7.69	0.079	3.751	1.06	0.078	3.411	0.99	**0.075**	**3.342**	**0.99**	0.083	4.410	0.99
03	0.131	4.822	2.47	0.089	1.889	7.45	0.051	1.493	1.17	0.058	0.602	1.01	**0.049**	**0.444**	**1.00**	0.047	0.608	1.00
04	0.126	1.899	1.08	0.037	0.846	6.42	0.037	0.816	0.50	**0.030**	**0.387**	0.44	0.030	0.428	**0.43**	0.028	0.549	0.33
05	0.148	5.563	3.35	0.155	10.630	9.75	0.089	6.352	2.40	**0.046**	1.285	**2.23**	0.046	**1.235**	2.23	0.056	1.644	2.17
06	0.142	3.376	1.55	0.066	1.984	7.30	0.044	**1.325**	0.63	0.032	1.576	**0.50**	**0.032**	1.569	0.50	0.031	1.467	0.45
07	0.170	5.347	6.41	0.258	12.558	12.51	0.120	5.371	5.58	**0.094**	2.731	**4.97**	0.098	**2.500**	5.15	0.073	2.132	4.18
08	0.144	8.508	3.49	0.088	3.902	8.91	0.053	2.908	2.49	0.048	2.373	**2.36**	**0.047**	**1.706**	2.36	0.047	2.454	2.31
09	0.151	4.546	1.71	0.054	2.027	6.76	0.052	2.307	0.74	0.043	1.244	**0.64**	**0.042**	**1.141**	0.64	0.044	1.385	0.64
10	0.148	6.540	2.88	0.119	8.302	8.53	0.066	4.576	1.78	0.058	3.789	**1.58**	**0.056**	**3.623**	1.60	0.057	2.615	1.37
train	0.168	6.407	2.69	0.173	8.301	8.59	0.103	3.955	1.73	**0.094**	2.782	**1.60**	0.096	**2.737**	1.61	0.100	3.193	1.52
test	0.146	7.246	2.97	0.085	4.237	8.34	0.055	3.060	1.96	0.048	2.359	**1.82**	**0.048**	**1.910**	1.83	0.048	2.234	1.76

Table 6. Quantitative comparison on the KITTI [21] dataset with KLT tracks [66]. We replace the Nistér-5pt [50] with the 8pt [42] algorithm to show more results. We also show, an approximation of the true error distance using reprojected points (this is excluded from being **bold** or underlined). While the reprojection approximation achieves the best results on almost all sequences, our methods are often not far behind. This emphasises, that our method is able to effectively learn covariances.

	8PT [42]			NEC [33]			NEC-LS			OURS SUPERVISED			OURS SELF-SUPERVISED			REPROJECTION		
Seq.	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t	RPE_1	RPE_n	e_t
00	0.216	11.650	3.40	0.132	12.483	3.20	0.116	8.728	**1.35**	0.114	**2.277**	1.38	**0.114**	2.522	1.38	0.113	2.363	1.28
01	0.246	8.080	3.83	0.539	22.857	1.55	0.082	6.378	1.00	0.060	5.811	0.99	**0.057**	**5.770**	**0.94**	0.054	5.997	0.81
02	0.188	12.003	2.06	0.093	7.594	1.76	0.069	4.050	1.01	0.066	**2.224**	**0.99**	**0.066**	2.237	1.00	0.065	2.679	0.95
03	0.167	8.308	3.42	0.090	3.863	3.31	**0.055**	3.754	1.12	0.059	2.239	1.13	0.057	**2.051**	**1.12**	0.054	2.394	1.07
04	0.160	2.682	1.45	0.040	0.486	0.81	0.041	**0.434**	0.49	0.038	1.041	**0.46**	**0.037**	0.808	0.46	0.027	0.526	0.30
05	0.198	9.236	4.56	0.119	11.779	3.65	0.062	12.437	2.50	0.055	**1.931**	**2.37**	**0.055**	1.949	2.40	0.053	2.123	2.02
06	0.193	5.244	2.89	0.059	6.901	1.43	0.050	6.634	0.76	0.042	**1.178**	0.70	**0.041**	1.242	**0.70**	0.035	0.964	0.58
07	0.231	7.086	8.86	0.185	4.402	8.67	0.112	**2.341**	6.69	**0.103**	2.772	**6.54**	0.109	3.715	6.63	0.120	3.434	4.82
08	0.183	10.423	4.21	0.081	8.284	3.66	0.056	7.004	2.50	**0.050**	**4.067**	2.46	0.050	4.118	**2.46**	0.048	3.623	2.30
09	0.185	5.485	2.29	0.053	1.646	1.43	0.052	1.553	0.71	0.049	1.317	0.71	**0.049**	**1.278**	**0.70**	0.048	1.160	0.69
10	0.198	8.960	4.09	0.167	9.264	4.43	0.064	4.787	1.79	**0.063**	**3.513**	**1.64**	0.065	3.821	1.65	0.060	2.404	1.21
train	0.203	10.051	3.54	0.141	10.127	2.97	0.082	6.910	1.72	0.077	**2.378**	**1.69**	**0.077**	2.505	1.69	0.076	2.606	1.44
test	0.186	9.023	3.74	0.089	6.917	3.28	0.056	5.353	1.96	**0.052**	**3.333**	1.91	0.053	3.408	**1.91**	0.050	2.839	1.73

Table 7. Full results on the KITTI [21] dataset with SuperPoint [12] keypoints. We replace the Nistér-5pt [50] with the 8pt [42] algorithm to show more results. We also show, an approximation of the true error distance using reprojected points (this is excluded from being **bold** or underlined). While the reprojection approximation achieves the best results on almost all sequences, our methods are often not far behind. This emphasises, that our method is able to effectively learn covariances.



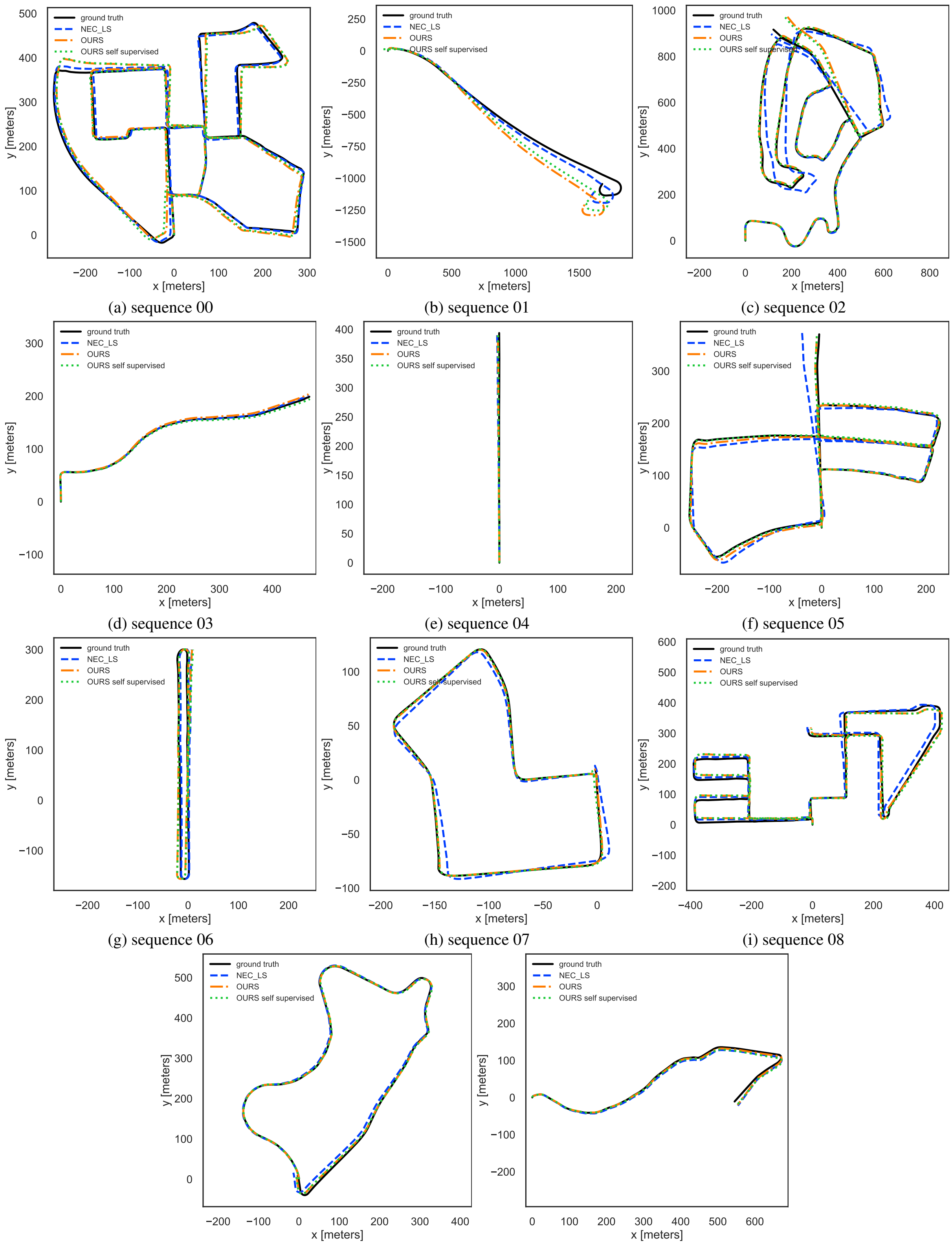
(a) sequence 00 (b) sequence 01 (c) sequence 02

(d) sequence 03 (e) sequence 04 (f) sequence 05

(g) sequence 06 (h) sequence 07 (i) sequence 08

(j) sequence 09 (k) sequence 10

Figure 19. Trajectory comparison for the KITTI visual odometry sequences for SuperPoint keypoints. Since we compare monocular methods, that cannot estimate the correct scale from a pair of images, we use the scale of the ground truth translations for visualization purposes.



(a) sequence 00 (b) sequence 01 (c) sequence 02

(d) sequence 03 (e) sequence 04 (f) sequence 05

(g) sequence 06 (h) sequence 07 (i) sequence 08

(j) sequence 09 (k) sequence 10

Figure 20. Trajectory comparison for the KITTI visual odometry sequences for KLT-tracks. Since we compare monocular methods, that cannot estimate the correct scale from a pair of images, we use the scale of the ground truth translations for visualization purposes.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org. 5

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 3

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 3

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

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017. 3

[6] Jesus Briales, Laurent Kneip, and Javier Gonzalez-Jimenez. A certifiably globally optimal solution to the non-minimal relative pose problem. In *CVPR*, 2018. 3

[7] M.J. Brooks, W. Chojnacki, D. Gawley, and A. van den Hengel. What value covariance information in estimating vision parameters? In *ICCV*, 2001. 1, 3, 7

[8] Keenan Burnett, David J Yoon, Angela P Schoellig, and Timothy D Barfoot. Radar odometry combining probabilistic estimation and unsupervised feature learning. *Robotics: Science and Systems*, 2021. 3

[9] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. 6, 8

[10] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32, 2016. 2

[11] 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. 1, 6, 8

[12] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018. 1, 2, 3, 4, 5, 7, 8, 15

[13] Justin Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*. PMLR, 2012. 4

[14] Leyza Baldo Dorini and Siome Klein Goldenstein. Unscented feature tracking. *Computer Vision and Image Understanding*, 115, 2011. 3

[15] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1, 2

[16] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *ECCV*, 2014. 2

[17] Kaveh Fathian, J Pablo Ramirez-Paredes, Emily A Doucette, J Willard Curtis, and Nicholas R Gans. Quest: A quaternion-based approach for camera motion estimation from minimal feature points. *IEEE Robotics and Automation Letters (RAL)*, 3, 2018. 3

[18] O.D. Faugeras and S. Maybank. Motion from point matches: multiplicity of solutions. In *Workshop on Visual Motion*, 1989. 1

[19] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 1981. 3

[20] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *ISPRS intercommission conference on fast processing of photogrammetric data*, 1987. 3

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 1, 6, 7, 8, 9, 15

[22] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2dnet: Learning accurate correspondences for sparse-to-dense feature matching. *arXiv preprint arXiv:2004.01673*, 2020. 3

[23] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19, 1997. 1

[24] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 1, 2

[25] Ganesh Iyer, Krishna Murthy Jatavallabhula, Gunshi Gupta, Madhava Krishna K, and Liam Paull. Geometric consistency for self-supervised end-to-end visual odometry. In *CVPR Workshops*, 2018. 3, 5

[26] Krishna Murthy Jatavallabhula, Ganesh Iyer, and Liam Paull. ∇ slam: Dense slam meets automatic differentiation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 3

[27] Kenichi Kanatani. For geometric inference from images, what kind of statistical model is necessary? *Systems and Computers in Japan*, 35, 2004. 3

[28] Kenichi Kanatani. Statistical optimization for geometric fitting: Theoretical accuracy bound and high order error analysis. *IJCV*, 80, 2008. 3

[29] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? In *ICCV*, 2001. 3

[30] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, 2015. 3

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2015. 6

[32] Laurent Kneip and Paul Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 7, 8, 11

[33] Laurent Kneip and Simon Lynen. Direct optimization of frame-to-frame rotation. In *ICCV*, 2013. 1, 2, 3, 7, 8, 15

[34] Laurent Kneip, Roland Siegwart, and Marc Pollefeys. Finding the exact rotation between two images independently of the translation. In *ECCV*, 2012. 1, 2, 3

[35] Erwin Kruppa. *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung*. Hölder, 1913. 2
[36] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *BMVC*, 2008. 3
[37] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2, 1944. 3
[38] Hongdong Li and Richard Hartley. Five-point motion estimation made easy. In *IEEE International Conference on Pattern Recognition (ICPR)*, 2006. 3
[39] John Lim, Nick Barnes, and Hongdong Li. Estimating relative camera motion from the antipodal-epipolar constraint. *IEEE TPAMI*, 32, 2010. 3
[40] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021. 1, 3
[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3
[42] HC Longuet-Higgins. Readings in computer vision: issues, problems, principles, and paradigms. *A computer algorithm for reconstructing a scene from two projections*, 1987. 1, 3, 11, 15
[43] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60, 2004. 3
[44] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981. 3
[45] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11, 1963. 3
[46] Jochen Meidow, Christian Beder, and Wolfgang Förstner. Reasoning with uncertain points, straight lines, and straight line segments in 2d. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64, 2009. 3
[47] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 3
[48] D Muhle, L Koestler, N Demmel, F Bernard, and D Cremers. The probabilistic normal epipolar constraint for frame-to-frame rotation optimization under uncertain feature positions. 2022. 1, 2, 3, 4, 6, 7, 8, 11
[49] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33, 2017. 6
[50] D. Nister. An efficient solution to the five-point relative pose problem. In *CVPR*, 2003. 2, 3, 7, 15
[51] D. Nistr, O. Naroditsky, and J. Bergen. Visual odometry. *CVPR*, 2004. 3, 8, 11
[52] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, Jing Dong, Brandon Amos, and Mustafa Mukadam. Theseus: A Library for Differentiable Nonlinear Optimization. *NeurIPS*, 2022. 5
[53] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 1, 2, 3
[54] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 3
[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015. 4, 7
[56] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1, 4
[57] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 3, 7, 8
[58] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021. 2, 3
[59] Sameer Sheorey, Shalini Keshavamurthy, Huili Yu, Hieu Nguyen, and Clark N Taylor. Uncertainty estimation for klt tracking. In *Asian Conference on Computer Vision*, 2014. 3
[60] R.M. Steele and C. Jaynes. Feature uncertainty arising from covariant image noise. In *CVPR*, 2005. 3
[61] Henrik Stewenius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60, 2006. 3
[62] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2
[63] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. *IJCV*, 9, 1991. 1, 3
[64] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *CVPR*, 2015. 3
[65] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, 1999. 2
[66] 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. 4, 5, 7, 15
[67] Lukas Von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. Gn-net: The gauss-newton loss for multi-weather relocalization. *IEEE Robotics and Automation Letters (RAL)*, 2020. 2, 3
[68] N. Yang, L. von Stumberg, R. Wang, and D. Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*, 2020. 3
[69] N. Yang, R. Wang, J. Stueckler, and D. Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *ECCV*, 2018. 3
[70] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 3

[71] Bernhard Zeisl, Pierre Georgel, Florian Schweiger, Eckehard Steinbach, and Nassir Navab. Estimation of location uncertainty for scale invariant feature points. In *BMVC*, 2009. 3

[72] Hongmou Zhang, Denis Grießbach, Jürgen Wohlfeil, and Anko Börner. Uncertainty model for template feature matching. In *Pacific-Rim Symposium on Image and Video Technology*, pages 406–420. Springer, 2017. 3