Learning Correspondence Uncertainty via Differentiable Nonlinear Least Squares

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

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.

Project Page: https://dominikmuhle.github.io/dnls_covs/
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).

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 inte-

consensus (RANSAC) [19] is a popular method to classify datapoints into inliers and outliers that can be easily inte-

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. DVS0 [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. VSLAM [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 positional 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. \( \mathbf{f} \)) denote vectors, whereas bold uppercase letters (e.g. \( \mathbf{S} \)) denote matrices. \( \mathbf{u} \in \mathbb{R}^{3 \times 3} \) represents the skew-symmetric matrix of the vector \( \mathbf{u} \in \mathbb{R}^3 \) such that the cross product between two vectors can be rewritten as a matrix-vector operation, i.e. \( \mathbf{u} \times \mathbf{v} = \hat{\mathbf{u}} \mathbf{v} \). The transpose is
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 $R \in SO(3)$ and a unit length translation $t \in \mathbb{R}^3$ ($||t|| = 1$ is imposed due to scale-invariance).

3.2. The Probabilistic Normal Epipolar Constraint

The asymmetric probabilistic normal epipolar constraint (PNEC) estimates the relative pose, give two images $I, I'$ of the same scene under the assumption of uncertain feature positions in the second image. A feature correspondence is given by $p_i, p_i'$ in the image plane, where the uncertainty of $p_i'$ is represented by the corresponding covariance $\Sigma_{2D,i}$. To get the epipolar geometry for the PNEC the feature points are unprojected using the camera intrinsics, giving unit length bearing vectors $f_i, f_i'$. The uncertainty of $f_i'$ is now represented by $\Sigma_{e,i}'$. Estimating the relative pose is done by minimizing the PNEC cost function as defined in [48].

For convenience we recap the energy function

$$E(R, t) = \sum_i \frac{e_i^2}{\sigma_i^2} = \sum_i \frac{|t^\top (f_i \times Rf_i')|^2}{t^\top f_i \Sigma_{e,i}' R^\top f_i'} ,$$

(1)

in our notation. As mentioned previously, this asymmetric PNEC in [48] only considers uncertainties $\Sigma_{e,i}'$ 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 $f_i$ and $f_i'$ to be gaussian, their covariance matrices denoted by $\Sigma_i$ and $\Sigma_{e,i}'$, respectively. The new variance can be approximated as

$$\sigma_{s,i}^2 = t^\top ((Rf_i')\Sigma_i (Rf_i')^\top + f_i'R\Sigma_{e,i}' R^\top f_i') t .$$

(2)

In the supplementary material 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

$$E_s(R, t) = \sum_i \frac{e_i^2}{\sigma_{s,i}^2} = \sum_i \frac{|t^\top (f_i \times Rf_i')|^2}{t^\top Rf_i' \Sigma_i R^\top f_i' R f_i'} ,$$

(3)

3.3. DNLS for Learning Covariances

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

$$\Sigma_{2D} = \arg \min_{\Sigma_{2D}} \mathcal{L} ,$$

(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

$$e_{rot} = \angle \tilde{R}^\top R$$

(5)

and

$$\mathcal{L}(\tilde{R}, R; \Sigma_{2D}) = e_{rot}$$

(6)

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

$$R = \arg \min_{\tilde{R}} E_s(R, t; \Sigma_{2D})$$

(7)

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

$$\frac{d\mathcal{L}}{d\Sigma_{2D}} = - \frac{\partial^2 E_s}{\partial \Sigma_{2D} \partial R^\top} \left( \frac{\partial^2 E_s}{\partial R \partial R^\top} \right)^{-1} e_{rot} .$$

(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].
Figure 4. Rotational error (a) and differences between the true residual variance $\tilde{\sigma}^2$ and the learned variance $\sigma^2$ (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 $\sigma^2$.

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 $F$ to learn the noise distributions of a keypoint detector. Given an image and a keypoint position, the network should predict the covariance of the noise $\Sigma_{2D,i} = F(I, p_i)$. The gradient $d\mathcal{L}/d\Sigma_{2D}$ 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

$$\mathcal{L}_{\text{sup}} = \varepsilon_{\text{rot}}$$

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 $\{I_1, I_2, I_3\}$ of images is given by

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

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

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

**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
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 $\sigma_{i,\text{norm}}^2$, given by

$$\sigma_{i,\text{norm}}^2 = \frac{N \cdot \sigma_i^2}{\sum_{j=0}^{N} \sigma_j^2}$$

over the training epochs, obtained at the ground truth relative pose. We compare the normalized error variance, as the scale of $\sigma^2$ is not observable from the gradient. The covariances that minimize the rotational error also approximate the residual uncertainty $\sigma^2$ 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 $\sigma_i^2$ dependence on both $\Sigma_{2D,i}$ and $\Sigma'_{2D,i}$, there is not a single unique solution. Secondly, the direction of the gradient is dependent on the translation between the images (see the supplementary material for more details). In this experimental setup, the information flow to the images is limited and we can only learn the true distribution for $\sigma^2$ 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 to learn the distribution for $\sigma^2$ but also $\Sigma'_{2D}$. 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,
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. 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 multi-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 supervised and self-supervised training can be found in [48].
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.

<table>
<thead>
<tr>
<th>Seq.</th>
<th>RPE(_1)</th>
<th>RPE(_e)</th>
<th>(e_t)</th>
<th>RPE(_1)</th>
<th>RPE(_e)</th>
<th>(e_t)</th>
<th>RPE(_1)</th>
<th>RPE(_e)</th>
<th>(e_t)</th>
<th>RPE(_1)</th>
<th>RPE(_e)</th>
<th>(e_t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1,01</td>
<td>0.501</td>
<td>71.87</td>
<td><strong>31.86</strong></td>
<td><strong>0.320</strong></td>
<td>39.50</td>
<td>43.12</td>
<td>0.387</td>
<td>52.92</td>
<td>46.31</td>
<td><strong>0.388</strong></td>
<td>56.52</td>
<td>46.82</td>
</tr>
<tr>
<td>V1,02</td>
<td>0.541</td>
<td>32.01</td>
<td><strong>20.36</strong></td>
<td><strong>0.389</strong></td>
<td>28.11</td>
<td>26.95</td>
<td>0.540</td>
<td>70.08</td>
<td>28.94</td>
<td>0.542</td>
<td>68.35</td>
<td>29.81</td>
</tr>
<tr>
<td>V1,03</td>
<td>0.660</td>
<td>27.39</td>
<td>25.00</td>
<td><strong>0.492</strong></td>
<td><strong>25.42</strong></td>
<td>31.06</td>
<td>0.552</td>
<td>76.72</td>
<td>31.58</td>
<td>0.555</td>
<td>78.14</td>
<td>32.25</td>
</tr>
<tr>
<td>V2,01</td>
<td>0.515</td>
<td>61.45</td>
<td>33.51</td>
<td>0.316</td>
<td>31.95</td>
<td>39.79</td>
<td>0.310</td>
<td>35.84</td>
<td>39.00</td>
<td>0.314</td>
<td>38.62</td>
<td>39.62</td>
</tr>
<tr>
<td>V2,02</td>
<td>0.545</td>
<td>43.73</td>
<td>22.24</td>
<td>0.396</td>
<td>25.48</td>
<td>32.21</td>
<td>0.369</td>
<td>26.96</td>
<td>25.36</td>
<td><strong>0.365</strong></td>
<td><strong>25.09</strong></td>
<td>25.81</td>
</tr>
<tr>
<td>V2,03</td>
<td>1.123</td>
<td><strong>36.71</strong></td>
<td><strong>28.77</strong></td>
<td>0.976</td>
<td>48.26</td>
<td>37.60</td>
<td><strong>0.939</strong></td>
<td><strong>107.11</strong></td>
<td><strong>36.74</strong></td>
<td>0.941</td>
<td><strong>100.73</strong></td>
<td><strong>36.71</strong></td>
</tr>
</tbody>
</table>

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**

5. Discussion and Limitations

Our experiments demonstrate the capability of our framework 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 the supplementary material.

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.
References


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


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

[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