Supplementary Material for
Minimal Solutions for Relative Pose with a Single Affine Correspondence

Banglei Guan[1], Ji Zhao*, Zhang Li[1], Fang Sun[1] and Friedrich Fraundorfer[2,3]

[1]College of Aerospace Science and Engineering, National University of Defense Technology, China

[2]Institute for Computer Graphics and Vision, Graz University of Technology, Austria

[3]Remote Sensing Technology Institute, German Aerospace Center, Germany

guanbanglei12@nudt.edu.cn zhaoji84@gmail.com zhangli_nudt@163.com

sunfang@nudt.edu.cn fraundorfer@icg.tugraz.at

1. Least-Squares Solution

Refer to Eq. (14) in the paper, by taking the partial derivatives with $\{x_i\}_{i=1}^4$ and $\{\lambda_i\}_{i=1}^2$ and set them to be zeros, we obtain an equation system with unknowns $\{x_i\}_{i=1}^4$ and $\{\lambda_i\}_{i=1}^2$:

$$\begin{aligned}
\frac{1}{2}\frac{\partial L}{\partial x_1} &= \sum_{i=1}^{3}\left[a_i^2 x_1 + a_i(b_i x_2 + c_i x_3 + d_i x_4)\right] + \lambda_1 x_1 = 0\\
\frac{1}{2}\frac{\partial L}{\partial x_2} &= \sum_{i=1}^{3}\left[b_i^2 x_2 + b_i(a_i x_1 + c_i x_3 + d_i x_4)\right] + \lambda_1 x_2 = 0\\
\frac{1}{2}\frac{\partial L}{\partial x_3} &= \sum_{i=1}^{3}\left[c_i^2 x_3 + c_i(a_i x_1 + b_i x_2 + d_i x_4)\right] + \lambda_2 x_3 = 0\\
\frac{1}{2}\frac{\partial L}{\partial x_4} &= \sum_{i=1}^{3}\left[d_i^2 x_4 + d_i(a_i x_1 + b_i x_2 + c_i x_3)\right] + \lambda_2 x_4 = 0\\
\frac{\partial L}{\partial \lambda_1} &= x_1^2 + x_2^2 - 1 = 0\\
\frac{\partial L}{\partial \lambda_2} &= x_3^2 + x_4^2 - 1 = 0
\end{aligned}$$

The above equation system contains 6 unknowns $\{x_1, x_2, x_3, x_4, \lambda_1, \lambda_2\}$, and the order is 2.

2. Relative Pose Estimation with Known Vertical Direction

We show the solution procedure of the coefficients β and γ. To derive the solution, we start by substituting Eq. (26) to Eqs. (27) and (28) in the paper. Six equations from the trace constraint Eq. (28), together with a equation from the singularity of the essential matrix Eq. (27), form a system of 7 polynomial equations in 2 unknowns $\{\beta, \gamma\}$, which has a maximum polynomial degree of 3. First, we stack 7 polynomial equations into a matrix form as

$$\mathbf{M}_1 \mathbf{v}_1 = 0, \tag{1}$$

where $\mathbf{v}_1 = [\beta^3, \beta^2\gamma, \beta^2, \beta\gamma^2, \beta\gamma, \beta, \gamma^3, \gamma^2, \gamma, 1]^T$, $\mathbf{M}_1$ is a 7×10 coefficient matrix.

Since there is a linear dependency between the elements of the essential matrix, *i.e.*, e_2, e_4, e_5 and e_6, the rank of the coefficient matrix $\mathbf{M}_1$ is only 6. By performing Gaussian elimination and row operations on the 6 linearly independent equations, we set up a new polynomial equation system as follows:

β^3	$\beta^2\gamma$	β^2	$\beta\gamma^2$	$\beta\gamma$	β	γ^3	γ^2	γ	1	
1						.	.	.	.	
	1					.	.	.	.	
		1				.	.	.	.	
			1			.	.	.	.	
				1		.	.	.	.	$\langle Q_a \rangle$
					1	.	.	.	.	$\langle Q_b \rangle$

where $Q_a = poly(\beta\gamma, \gamma^3, \gamma^2, \gamma, 1)$ and $Q_b = ploy(\beta, \gamma^3, \gamma^2, \gamma, 1)$ represent the polynomial in the fifth and sixth rows, respectively.

In order to eliminate the monomial $\beta\gamma$, we multiply Q_b with γ and subtract it from Q_a:

$$Q_c = \gamma Q_b - Q_a = poly(\gamma^4, \gamma^3, \gamma^2, \gamma, 1) \tag{2}$$

Now, we get an up to degree 4 polynomial in γ: Q_c. The unknown γ has at most 4 solutions and can be computed as the eigenvalues of the companion matrix of Q_c. Then the corresponding solution for the unknown β is obtained directly by substituting γ into Q_b.

3. Experiments

3.1. Efficiency Comparison

We evaluate the run-times of our solvers and the comparative solvers on an Intel(R) Core(TM) i7-8550U 1.80GHz

*Corresponding author.

using MATLAB. All algorithms are implemented in Matlab, except that the `5pt-Nister` method is implemented in C by using mex file. All timings are averaged over 10000 runs. Table 1 summarizes the run-times for the planar motion estimation algorithms[1]. The run-times of the methods `1AC-Voting` and `1AC-CS` are same and quite low, because both methods use the same solver and the computational complexity is mainly about computing the eigenvector of the matrix. For the methods `1AC-LS` and `1AC-UnknownF`, the high run-times are due to the complexity of the Gröbner basis solution.

Methods	6pt-Kukelova [3]	2pt-Choi [2]	**1AC-CS**	**1AC-LS**	**1AC-Voting**	**1AC-UnknownF**
Timings	0.405	0.098	**0.007**	0.120	**0.007**	0.196

Table 1. Run-time comparison of planar motion estimation algorithms (unit: ms).

Table 2 summarizes the run-times for the motion estimation algorithms with known vertical direction. The run-time of the `3pt-Saurer` method is higher than the `1AC method` method due to the complexity of the Gröbner basis solution. Since the mex file is used, the run-time of the `5pt-Nister` method is low. The run-time of the `1AC method` method is significantly lower than the `2AC-Barath` method, because the essential matrix between two views is simplified when the common direction of rotation is known, and we use a low-complexity approach to solve the essential matrix as shown in Section 2.

Methods	5pt-Nister [4]	3pt-Sweeney [6]	3pt-Saurer [5]	2pt-Saurer [5]	2AC-Barath [1]	**1AC method**
Timings	0.118	0.174	2.066	**0.097**	65.101	1.212

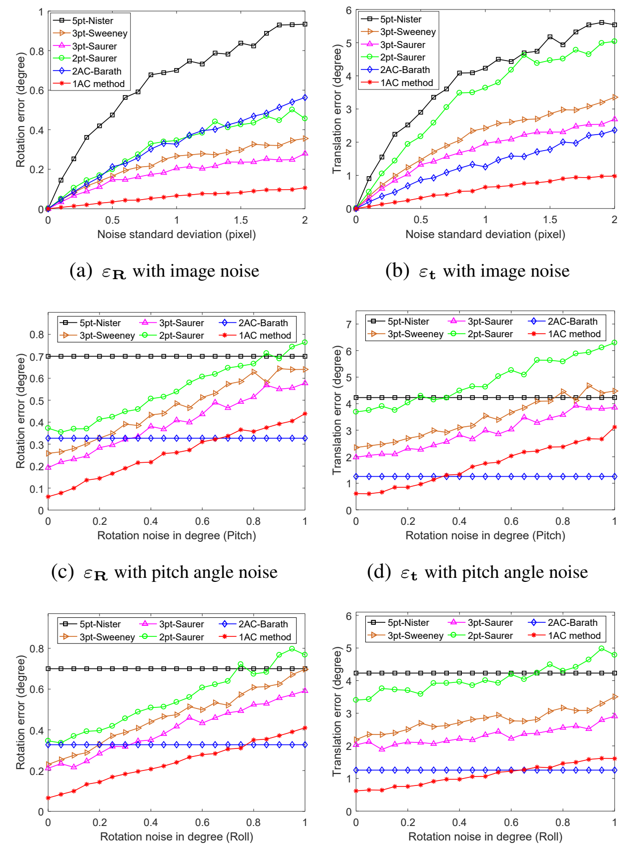
Table 2. Run-time comparison of motion estimation algorithms with known vertical direction (unit: ms).

3.2. Motion with Known Vertical Direction

In this section we show the performance of the proposed `1AC method` under forward and sideways motion. Figure 1 shows the performance of the proposed method under forward motion. Figure 2 shows the performance of the proposed method under sideways motion.

3.3. Visual Odometry

Here we show more trajectories for the experiments with `KITTI` dataset[2], see Figure 3. It shows that the proposed `1AC method` method has the smallest ATE among all the compared trajectories.



(a) $\varepsilon_{\mathbf{R}}$ with image noise (b) $\varepsilon_{\mathbf{t}}$ with image noise

(c) $\varepsilon_{\mathbf{R}}$ with pitch angle noise (d) $\varepsilon_{\mathbf{t}}$ with pitch angle noise

(e) $\varepsilon_{\mathbf{R}}$ with roll angle noise (f) $\varepsilon_{\mathbf{t}}$ with roll angle noise

Figure 1. Rotation and translation error under forward motion (unit: degree). (a)(b): vary image noise with perfect IMU data. (c)∼(f): vary IMU angle noise and fix the image noise as 1.0 pixel standard deviation. The left column reports the rotation error. The right column reports the translation error.

References

[1] Daniel Barath and Levente Hajder. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, 27(11):5328–5337, 2018.

[2] Sunglok Choi and Jong-Hwan Kim. Fast and reliable minimal relative pose estimation under planar motion. *Image and Vision Computing*, 69:103–112, 2018.

[3] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4912–4921, 2017.

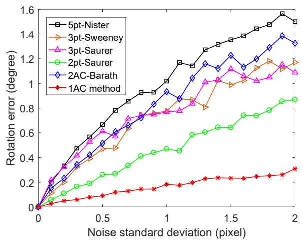
[4] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):0756–777, 2004.

[5] Olivier Saurer, Pascal Vasseur, Rémi Boutteau, Cédric Demonceaux, Marc Pollefeys, and Friedrich Fraundorfer. Homography based egomotion estimation with a common direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):327–341, 2016.

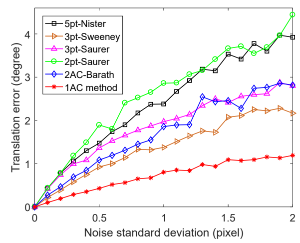
[6] Chris Sweeney, John Flynn, and Matthew Turk. Solving for relative pose with a partially known rotation is a quadratic eigenvalue problem. In *International Conference on 3D Vision*, 2014.

[1] Note that the run-times of the methods `5pt-Nister` and `2AC-Barath` are showed in Table 2.

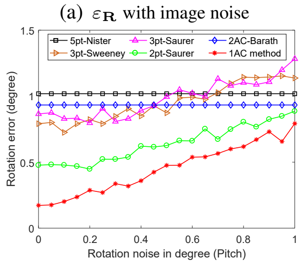
[2] Both `ORB-SLAM2` and our monocular visual odometry fail to produce a valid result for sequence 01, because it is a highway with few tractable close objects.



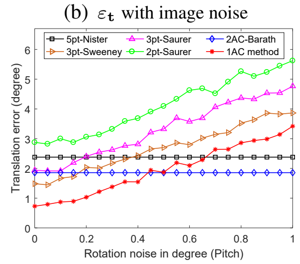
(a) $\varepsilon_{\mathbf{R}}$ with image noise



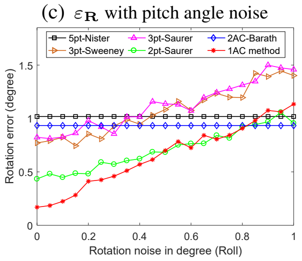
(b) $\varepsilon_{\mathbf{t}}$ with image noise



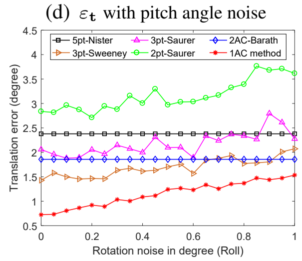
(c) $\varepsilon_{\mathbf{R}}$ with pitch angle noise



(d) $\varepsilon_{\mathbf{t}}$ with pitch angle noise

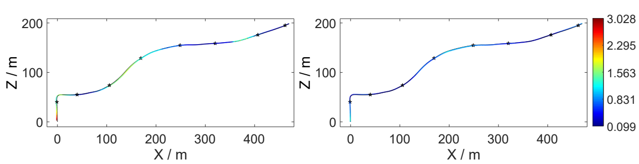


(e) $\varepsilon_{\mathbf{R}}$ with roll angle noise

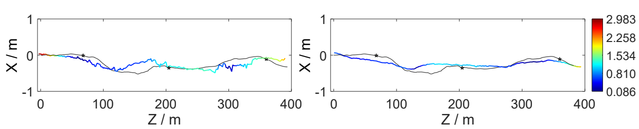


(f) $\varepsilon_{\mathbf{t}}$ with roll angle noise

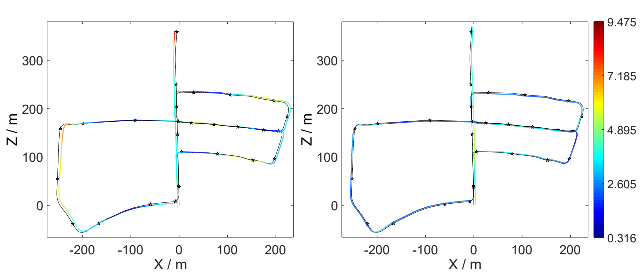
Figure 2. Rotation and translation error under sideways motion (unit: degree). (a)(b): vary image noise with perfect IMU data. (c)∼(f): vary IMU angle noise and fix the image noise as 1.0 pixel standard deviation. The left column reports the rotation error. The right column reports the translation error.



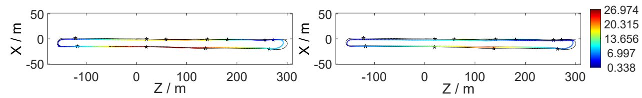
(a) Seq.03



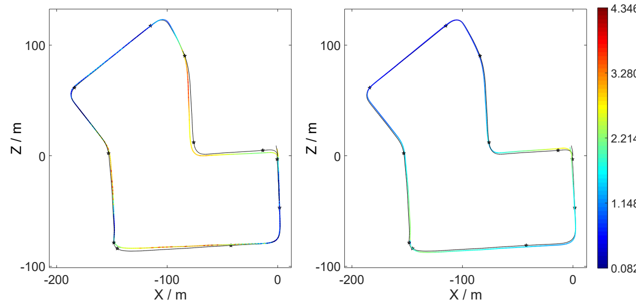
(b) Seq.04



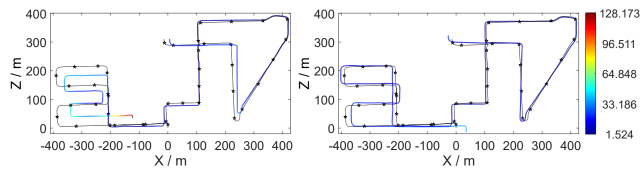
(c) Seq.05



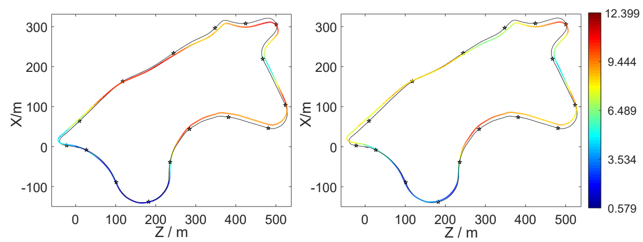
(d) Seq.06



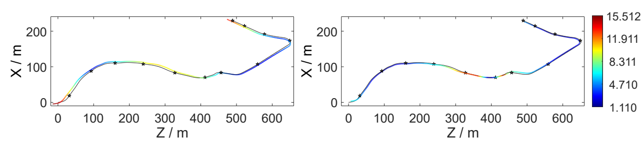
(e) Seq.07



(f) Seq.08



(g) Seq.09



(h) Seq.10

Figure 3. Estimated visual odometry trajectories. The left column reports the results of `ORB-SLAM2`. The right column reports the results of our monocular visual odometry. Colorful curves are estimated trajectories, and black curves with stars are ground truth trajectories. Best viewed in color.