Supplementary material: Leveraging scale- and orientation-covariant features for planar motion estimation

Marcus Valtonen Örnhag and Alberto Jaenal

Ericsson Research, Sweden
{marcus.valtonen.ornhag,alberto.jaenal.galvez}@ericsson.com

1 Introduction

In the supplementary material we include parts that were left out from the main paper due to page limitations:

- Leveraging SIFT and affine constraints in optimal planar motion estimation for refinement, by incorporating the novel constraint in [5].
- Histogram voting – affine features vs SIFT features.
- The effect on the number of inliers when using SIFT features.

Equations with roman numerals refer to this document, otherwise they refer to the main paper.

2 Extension to least-squares optimal planar motion solvers

In [5] a non-minimal optimal relative planar motion solver was proposed. They use point-based correspondences, yielding a matrix equation of the form (17) utilizing the epipolar constraint (8) alone. In this case, N point correspondences creates a matrix of size $N \times 4$. They proceed by devising an algorithm to minimize the least squares error

$$\min_{\boldsymbol{e} \in \mathcal{C}} \|\boldsymbol{M}\boldsymbol{e}\|_2^2, \tag{i}$$

where $\mathcal{C}$ is the set of vectors $\boldsymbol{e} \in \mathbb{R}^4$ of unit length fulfilling the trace constraint (7). It is straight-forward to generalize this to SIFT and affine correspondences. Using N SIFT correspondences the matrix $\boldsymbol{M}$ has size $2N \times 4$, due to the added SIFT constraint (16) and analogously, with affine correspondences, a matrix of size $3N \times 4$ is obtained, due to the additional two constraints

$$a_1 v_i e_1 + (a_3 u_i + v_j) e_2 + a_3 e_3 = 0, \tag{ii}$$

$$(a_2 v_i + u_j) e_1 + a_4 u_i e_2 + a_4 e_3 + e_4 = 0, \tag{iii}$$

reported in [4]. No additional changes are needed to the algorithm in order to make it utilize the extra information embedded in SIFT descriptors and affine correspondences.

2.1 Experiments

Synthetic data

Sensitivity to noise. We evaluate the non-minimal solver with the modifications proposed in Sec. 2. The results are shown in Fig. 1. Unsurprisingly, the point-based solver, which uses less information, performs worse than the competing methods, and initially the affine correspondences give better results than using SIFT features. This is likely due to the fact that three instead of two equations are used for affine correspondences; however, the differences quickly become negligible and around 15 correspondences, the methods perform almost identical.

Nonplanar motion. The results for the non-minimal solvers are shown in Fig. 2. Similar to the previous comparison, both the affine and SIFT-based approach quickly converge, and only initially, there is a slight favor for the affine correspondences.

Timings. As the non-minimal solvers simply scale with the size of the coefficient matrix $\boldsymbol{M}$, *cf*. (i), they are omitted.

Real data We now show the results obtained on the Malaga Urban dataset [2]. Here we use the default settings of GC-RANSAC [1] as in the main paper, with the exception that the non-minimal solver is used for refinement. The results shown in Fig. 3 compare the 5 PT solver from Nister using Stewenius algorthm as the non-linear refinement with the three planar methods using Hajder algotihm [5] as refinement. Note that, instead of using the 95 % of the data for refinement as in [5], we employ the GC-RANSAC default setting of 5 % of the data for fair comparison. In general, Hajder algorithm performs worse than the Stewenius solver used in the main paper, which is the main reason it was not included. We believe that the nonplanar differences, alignment of cameras, etc., do not work in favour of a pure planar motion model, and that some non-linear refinement extending the results outside the model is necessary for it to work well in practice. From what we can see at the experiment, however, is that our method is superior in execution time to the the competing methods, which is the general trend. This excludes the processing time of features, which would—in line with the main paper—further distinguish our method from the affine-based solver.

2.2 Histogram voting

In the main paper, we mentioned histogram voting and argued that it can be applied to the 1 AC solver [4] and the proposed 1 SIFT solver. Since there are two parameters we use histogram voting on the rotational component θ alone, and then select the corresponding values for ϕ. This results in a nice Gaussian distribution on the KITTI dataset [3], see Fig. 4, and we can simply use the median and not explicitly compute the histogram, as proposed in [6].

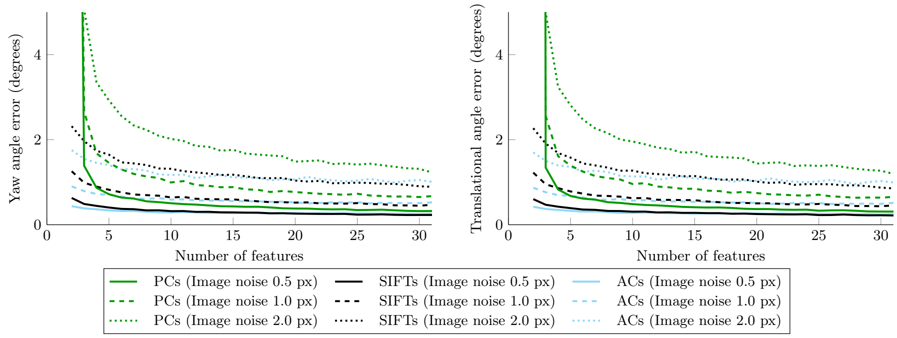

Fig. 1: *Sensitivity to noise.* Relative planar pose estimation in a synthetic environment. The angular errors (veritcal axis; in degrees) of the estimated rotation and translation as a function of the number of features (point-based, SIFTs, and affine correspondences) averaged over 5 000 runs per number of features used (horizontal axis). The state-of-the-art solver in [5] is used with modifications proposed in Sec. 2. Better seen in color.

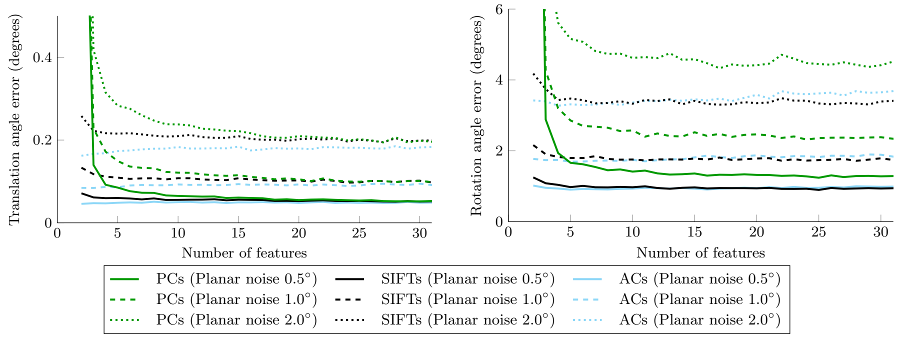

Fig. 2: *Nonplanar motion.* Relative planar pose estimation in a synthetic environment. The median angular error (veritcal axis; in degrees) of the estimated rotation is shown as functions function of the number of features used. The state-of-the-art solver in [5] is used with modifications proposed in Sec. 2. Better seen in color.

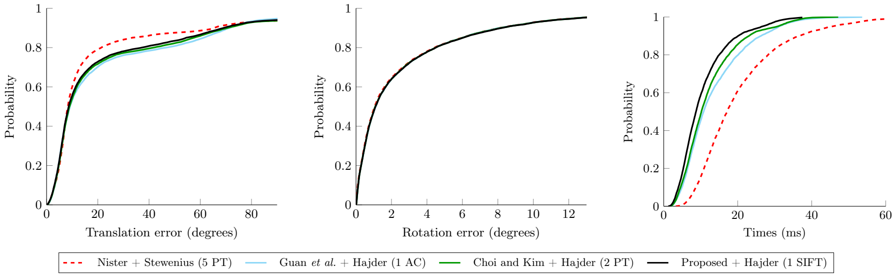

Fig. 3: *Malaga experiments.* Cumulative density functions deploying the performance of four different solvers in terms of angular error and GC-RANSAC processing time (excluding pre-processing time) for the Malaga dataset. A method being accurate (or fast) is equivalent to its curve being on the left side of the plot.

Real data We compare the proposed 1 SIFT solver and the 1 AC solver using histogram voting on the KITTI dataset is shown in Fig. 5. For fair comparison, the solution is refined using the Stewenius algorithm as in the main paper. In general, both algorithms have a similar performance in terms of accuracy, but our method outperforms Guan solver in computational terms, being around 3× faster—excluding the pre-processing times, which leads to an even more significant improvement. However, the accuracy obtained by the histogram voting framework in both cases is far from the one achieved by the traditional GC-RANSAC pipeline. On the other hand, with median times around 0.13 and 0.37 milliseconds respectively for our and Guan solvers, we report that this framework is around 40× faster.

The accuracy is not in line with the results in [4]; however, they do not supply code, nor detail their implementation, and we have not been able to reproduce their results. It is fair to believe, however, that if their implementation differs and is indeed superior to our simple histogram voting implementation, that our SIFT-based solver would perform equally well and in less time when integrated.

3 Effect on the number of inliers when using SIFT features

One might think that the number of inliers is smaller when the scale and orientation are taken into consideration; however, this is not the case, as is seen in Tab. 1. E.g., comparing our proposed solver to Choi's solver there is not a significant difference. This is primarily because we use the same metric (distance to the epipolar line) to separate inliers from outliers, and these are only taking the pixel coordinates into account.

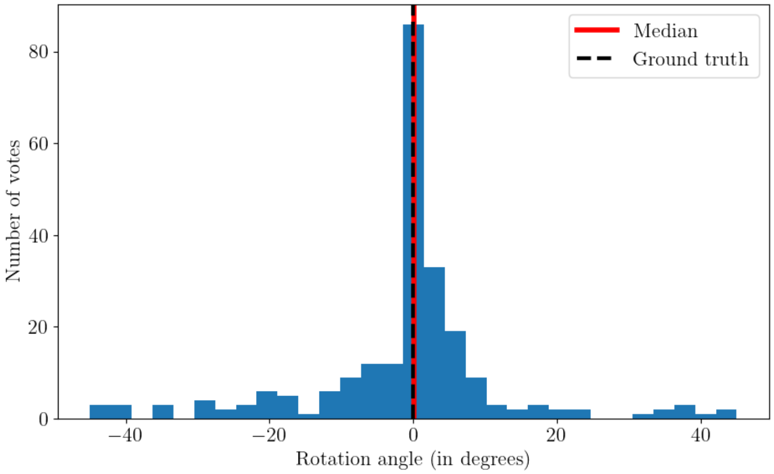

Fig. 4: *Histogram voting.* Example of histogram obtained on the KITTI dataset with the proposed SIFT-based algorithm.

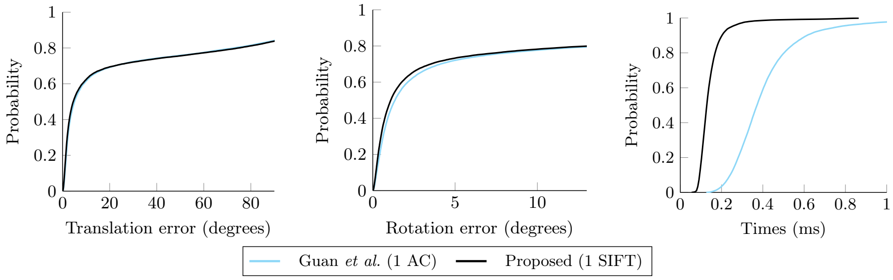

Fig. 5: *Histogram voting experiments.* Cumulative density functions deploying the performance of the single correspondence solvers within a histogram voting framework, in terms of angular error and processing time (excluding pre-processing time) for the KITTI dataset. A method being accurate (or fast) is equivalent to its curve being on the left side of the plot.

Table 1: Inliers (%) on the KITTI dataset.

Seq	00	01	02	03	04	05	06	07	08	09	10	All
Nister (5PC)	72.26	73.72	68.15	85.40	75.42	73.60	63.35	79.94	72.69	63.14	68.12	72.34
Choi (2PC)	72.20	73.50	68.08	85.40	75.52	73.65	63.11	79.85	72.69	63.22	68.12	72.30
Guan (1AC)	77.02	75.96	71.70	85.47	81.14	78.74	71.12	81.54	77.09	68.00	71.13	76.26
Our (1SIFT)	72.25	73.55	68.09	85.41	75.51	73.43	63.20	80.13	72.62	63.22	68.12	72.32

4 Code for generating the elimination ideal

In order to generate Eq. (16) of the main paper, the following Macaulay2 code was used:

```
KK = ZZ / 30097;
R = KK[e1,e2,e3,e4,u1,v1,u2,v2,q,s1,s2,c1,c2,a1,a2,a3,a4,MonomialOrder=>GRevLex];
eq1 = e1*v1*a1 + e2*u1*a3 + e2*v2 + e3*a3;
eq2 = e1*v1*a2 + e1*u2 + e2*u1*a4 + e3*a4 + e4;
eq3 = q^2 - a1*a4 + a2*a3;
eq4 = -q*s2 + s1*a4 + c1*a3;
eq5 = -q*c2 + s1*a2 + c1*a1;
I = ideal {eq1,eq2,eq3,eq4,eq5};
I = eliminate(I,{a1,a2,a3,a4});
<< gens I << "\n"
exit();
```

References

1. Barath, D., Matas, J.: Graph-cut ransac. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
2. Blanco-Claraco, J.L., Moreno-Duenas, F.A., González-Jiménez, J.: The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. The International Journal of Robotics Research (IJRR) **33**(2), 207–214 (2014)
3. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3354–3361 (June 2012)
4. Guan, B., Zhao, J., Li, Z., Sun, F., Fraundorfer, F.: Minimal solutions for relative pose with a single affine correspondence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
5. Hajder, L., Barath, D.: Least-squares optimal relative planar motion for vehicle-mounted cameras. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 8644–8650 (2020)
6. Scaramuzza, D.: 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. International Journal of Computer Vision (IJCV) **95**(1), 74–85 (October 2011)