— Supplementary Material —
Absolute Pose from One or Two Scaled and Oriented Features

Jonathan Ventura[1] Zuzana Kukelova[2] Torsten Sattler[3] Dániel Baráth[4]
[1] Department of Computer Science & Software Engineering, Cal Poly, San Luis Obispo
[2] Visual Recognition Group, Faculty of Electrical Engineering, Czech Technical University in Prague
[3] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
[4] Computer Vision and Geometry Group, ETH Zürich

1. Outline

Our supplementary material provides additional derivations and experiments to support the material in the main paper. In Sec. 2, we write out in full the absolute pose constraints from scale and orientation features (as mentioned in Sec. 3.4 in the main paper). In Sec. 3, we analyze robust estimation performance, noise interactions (as mentioned in Sec. 4.1 in the main paper), and degenerate configurations through experiments on synthetic data. In Sec. 4 we discuss affine feature extraction. In Sec. 5 we discuss inlier scoring and pose refinement.

2. Systems of equations for minimal solvers

2.1. P2ORI

The general absolute pose problem has six DOFs (three for rotation, three for translation), and thus, we need a system of six independent equations to solve it. In our case, each correspondence provides five constraints in total: two from the point projection (Eqs. 6 and 7 in the main paper) and three from the scale and orientation (Eqs. 2, 3, and 5 in the main paper). Thus, one correspondence is insufficient and we need two correspondences to solve the full 6DoF problem.

Two correspondences provide us with ten equations of which six are sufficient to solve the problem. It is natural to use the four point projection constraints (Eqs. 6 and 7), since they are not affected by other types of noise. We need one additional constraint per observation to remove the two remaining DOFs, which could be Eqs. 2, 3, 4 or 5. We chose Eq. 4 because Eqs. 2 and 3 require both orientation and scale and Eq. 5 is quadratic and thus would result in a more complex solver. Here, we write out a complete derivation of the constraint on the orientations (Eq. 4).

For ease of reading, we repeat here the form of the affine matrix from Eq. 1 in the main paper:

$$\mathsf{A} = \frac{d}{m}(\mathsf{R}_{1:2,1:2}(\mathbf{n}_{\text{ref}}^T\tilde{\mathbf{p}}_{\text{ref}}) - (\mathsf{R}_{1:2,:}\tilde{\mathbf{p}}_{\text{ref}})\mathbf{n}_{\text{ref}}{}_{1:2}^T - \mathbf{p}_{\text{query}}(\mathsf{R}_{3,1:2}(\mathbf{n}_{\text{ref}}^T\tilde{\mathbf{p}}_{\text{ref}}) - (\mathsf{R}_{3,:}\tilde{\mathbf{p}}_{\text{ref}})\mathbf{n}_{\text{ref}}{}_{1:2}^T)) \quad , \tag{1}$$

where $\tilde{\mathbf{p}}_{\text{ref}} = [\mathbf{p}_{\text{ref}}^T \; 1]^T$, $\mathbf{n}_{\text{ref}} = \mathsf{R}_{\text{ref}}\mathbf{n}$, and $m = \mathbf{n}_{\text{ref}}^T\tilde{\mathbf{p}}_{\text{ref}}(d(\mathsf{R}_{3,:}\tilde{\mathbf{p}}_{\text{ref}}) + t_3)$.

We introduce the following substitutions:

$$b = (\mathbf{n}_{\text{ref}}^T\tilde{\mathbf{p}}_{\text{ref}}), \tag{2}$$

$$\mathbf{p}'_{\text{ref}} = \mathsf{R}\tilde{\mathbf{p}}_{\text{ref}}, \tag{3}$$

to rewrite Eq. (1) in a condensed form:

$$\mathsf{A} = \frac{d}{m}(b\mathsf{R}_{1:2,1:2} - \mathbf{p}'_{\text{ref}_{1:2}}\mathbf{n}_{\text{ref}}{}_{1:2}^T - \mathbf{p}_{\text{query}}(b\mathsf{R}_{3,1:2} - p'_{\text{ref}_3}\mathbf{n}_{\text{ref}}{}_{1:2}^T)) \quad , \tag{4}$$

and $m = b(dp'_{\text{ref}_3} + t_3)$.

Recall that a_1, a_2, a_3, a_4 are the elements of A in row-major order. Now we have

$$a_1 = \frac{d}{m}(br_{11} - p'_{\text{ref}_1}n_{\text{ref}_1} - p_{\text{query}_1}(br_{31} - p'_{\text{ref}_3}n_{\text{ref}_1})), \tag{5}$$

$$a_2 = \frac{d}{m}(br_{12} - p'_{\text{ref}_1}n_{\text{ref}_2} - p_{\text{query}_1}(br_{32} - p'_{\text{ref}_3}n_{\text{ref}_2})), \tag{6}$$

$$a_3 = \frac{d}{m}(br_{21} - p'_{\text{ref}_2}n_{\text{ref}_1} - p_{\text{query}_2}(br_{31} - p'_{\text{ref}_3}n_{\text{ref}_1})), \tag{7}$$

$$a_4 = \frac{d}{m}(br_{22} - p'_{\text{ref}_2}n_{\text{ref}_2} - p_{\text{query}_2}(br_{32} - p'_{\text{ref}_3}n_{\text{ref}_2})). \tag{8}$$

The system of equations for the P2ORI solver combines the projection constraints (Eqs. 6 and 7) with Eq. 4, re-

written here for convenience:

$$c_{\text{ref}} s_{\text{query}} a_1 + s_{\text{ref}} s_{\text{query}} a_2 - c_{\text{ref}} c_{\text{query}} a_3 - c_{\text{query}} s_{\text{ref}} a_4 = 0. \tag{9}$$

Plugging Eqs. (5) to (8) into Eq. (9) gives

$$\begin{aligned} & c_{\text{ref}} s_{\text{query}} (br_{11} - p'_{\text{ref}_1} n_{\text{ref}_1} - p_{\text{query}_1} (br_{31} - p'_{\text{ref}_3} n_{\text{ref}_1})) \\ & + s_{\text{ref}} s_{\text{query}} (br_{12} - p'_{\text{ref}_1} n_{\text{ref}_2} - p_{\text{query}_1} (br_{32} - p'_{\text{ref}_3} n_{\text{ref}_2})) \\ & - c_{\text{ref}} c_{\text{query}} (br_{21} - p'_{\text{ref}_2} n_{\text{ref}_1} - p_{\text{query}_2} (br_{31} - p'_{\text{ref}_3} n_{\text{ref}_1})) \\ & - c_{\text{query}} s_{\text{ref}} (br_{22} - p'_{\text{ref}_2} n_{\text{ref}_2} - p_{\text{query}_2} (br_{32} - p'_{\text{ref}_3} n_{\text{ref}_2})) \\ & = 0, \end{aligned} \tag{10}$$

where we have multiplied both sides by $\frac{m}{d}$ to make the equation linear in the unknown query rotation matrix and translation vector and remove d which is common to all terms. After parameterizing the rotation matrix with the Cayley parameterization, the equation becomes non-linear.

2.2. UP1SIFT

In the gravity-aware case, we assume that we have a measurement of the current gravity direction in the query camera's coordinate system. Assuming that the Y-axis of the world coordinate system is aligned with gravity, from the current measurement of gravity, we can determine a rotation R_{XZ} which rotates the Y-axis of the world coordinate system to align with the observation of gravity in the camera's coordinate system. The remaining unknown rotation R_Y is a rotation about the gravity direction, and thus the rotation is reduced to a single DOF. The complete query camera rotation can be written as $\mathsf{R}_{\text{query}} = \mathsf{R}_Y \mathsf{R}_{XZ}$.

Since we have four DOF and a single observation, we need to add two constraints to the point projection constraints (Eqs. 6 and 7), which could be any combination of Eqs. 2, 3, 4, or 5. We opted not to use Eq. 5, since it is quadratic in A, leaving Eq. 2, 3, or 4. We chose Eqs. 2, 3, although the combinations of Eqs. 2 and 4 or Eqs. 3 and 4 would likely lead to similar solvers.

The system of equations for the UP1SIFT solver combines the point projection constraints (Eqs. 6, 7) with Eqs. 2 and 3, rewritten here for convenience:

$$a_1 c_{\text{ref}} + a_2 s_{\text{ref}} - q c_{\text{query}} = 0, \tag{11}$$
$$a_3 c_{\text{ref}} + a_4 s_{\text{ref}} - q s_{\text{query}} = 0, \tag{12}$$

Plugging Eqs. (5) to (8) into Eqs. (11) and (12) gives

$$\begin{aligned} & d c_{\text{ref}} (br_{11} - p'_{\text{ref}_1} n_{\text{ref}_1} - p_{\text{query}_1} (br_{31} - p'_{\text{ref}_3} n_{\text{ref}_1})) \\ & + d s_{\text{ref}} (br_{12} - p'_{\text{ref}_1} n_{\text{ref}_2} - p_{\text{query}_1} (br_{32} - p'_{\text{ref}_3} n_{\text{ref}_2})) \\ & - mqc_{\text{query}} = 0, \end{aligned} \tag{13}$$

$$\begin{aligned} & d c_{\text{ref}} (br_{21} - p'_{\text{ref}_2} n_{\text{ref}_1} - p_{\text{query}_2} (br_{31} - p'_{\text{ref}_3} n_{\text{ref}_1})) \\ & + d s_{\text{ref}} (br_{22} - p'_{\text{ref}_2} n_{\text{ref}_2} - p_{\text{query}_2} (br_{32} - p'_{\text{ref}_3} n_{\text{ref}_2})) \\ & - mqs_{\text{query}} = 0, \end{aligned} \tag{14}$$

where we have multiplied both sides of the equations by m to make them linear in the unknown query rotation matrix and translation vector. After parameterizing the rotation matrix with the tangent half-angle parameterization, the equations become non-linear.

3. Extra synthetic data experiments

3.1. Robust estimation

To evaluate the efficiency of the various solvers in robust estimation in a controlled experiment, we tested each solver inside MSAC [3, 8] and Locally Optimized MSAC (LO-MSAC) [2, 4] on synthetic data problems with random outliers. LO-MSAC helps mitigate noise in the observations by using non-linear optimization to refine minimal sample solutions and grow the inlier set obtained from a minimal sample. We used the implementations of MSAC and LO-MSAC provided in RansacLib [7].

We increased the outlier rate from 0 to 0.9 and calculated the average timing of each method at each setting. Outliers were introduced by setting a proportion of the observations to random values. We used our default noise settings of 1 deg point noise, 1 deg normal noise, 1 deg orientation noise, 0.1 log scale noise, and 0.5 deg gravity noise.

The results are shown in Fig. 1. With vanilla MSAC, UP1SIFT is faster than all other solvers past an outlier ratio of about 0.35, but P2ORI is slower than the other solvers across all outlier ratios due to noise sensitivity. However, note that the vanilla MSAC experiment is only meant to provide a theoretical analysis of solver performance; any modern practical application would use LO-MSAC or more sophisticated variants such as GC-RANSAC [1] for best performance.

When using LO-MSAC to mitigate noise sensitivity, UP1SIFT is faster than all other methods past an outlier ratio of 0.2, and P2ORI is faster than P3P past an outlier ratio of about 0.4. We did not test GC-RANSAC [1] because the random synthetic data does not exhibit spatial coherence and thus the graph cut method would not be beneficial in these experiments.

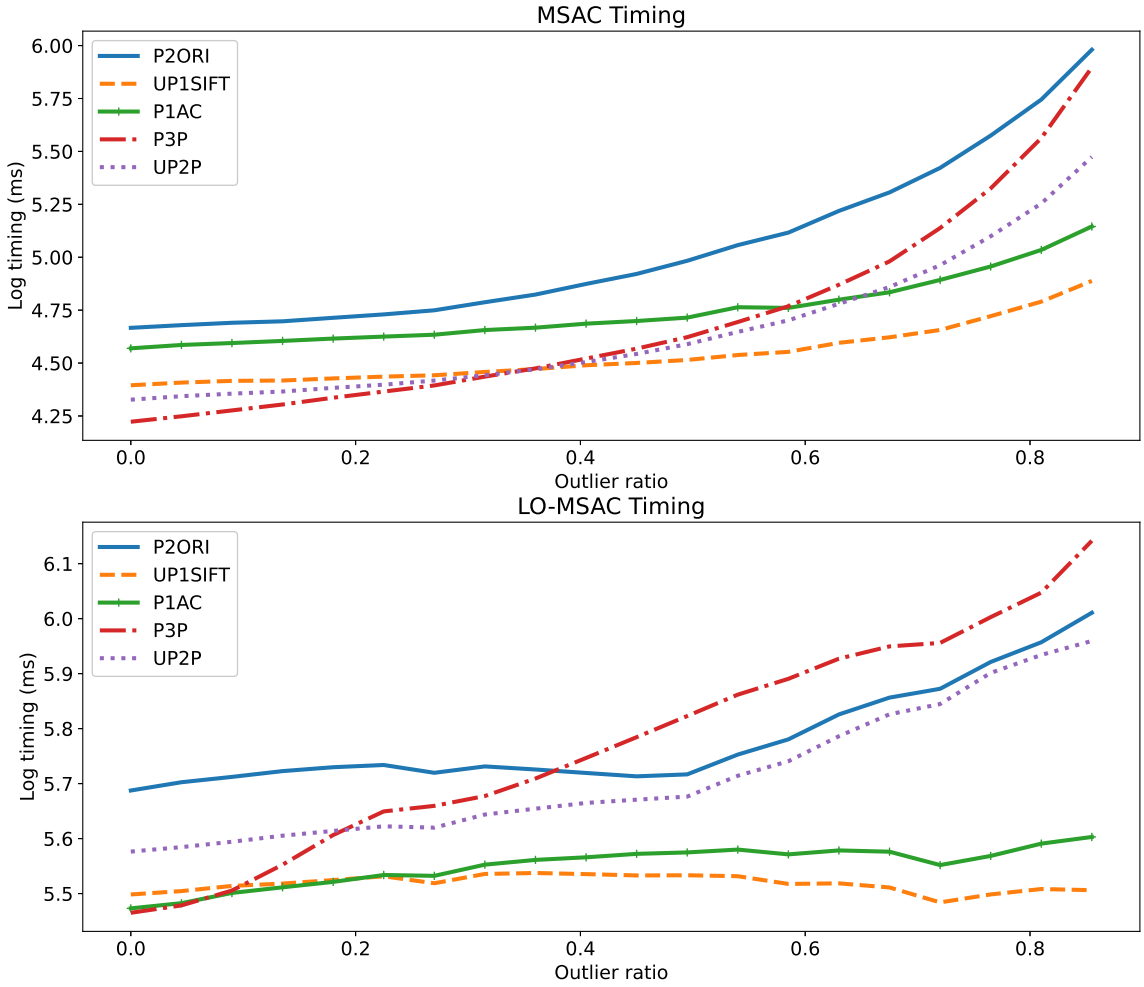

Figure 1. Average log timing (ms) for MSAC (*top*) and LO-MSAC (*bottom*) with various solvers and increasing outlier ratio. For all tests we used our default noise settings: 1 deg point noise, 1 deg normal noise, 1 deg orientation noise, 0.1 log scale noise, and 0.5 deg gravity noise.

3.2. Noise interaction

To explore interactions between noise types, we simultaneously varied pairs of noise types in synthetic data experiments. The results are shown in Figs. 2 to 4. The conclusions are largely the same as the single-noise experiments; namely, that P2ORI is most sensitive to orientation noise, and UP1SIFT is most sensitive to orientation noise in the rotation estimate and scale noise in the position estimate.

Because of the scale of the color bars, the increase in error with increasing point noise is sometimes not obvious in the plots. However, the error does indeed increase with point noise for all solvers, as can be more clearly seen in the 1D noise plots in Fig. 2 in the main paper.

It is clear that high noise in two factors will affect the precision of solvers working with these measurements. However, as shown in our real data experiments (Sec. 4.2 in the main paper), our solvers outperform other point/affine solvers in real noise settings.

3.3. Degenerate configurations

In the main paper (Secs. 3.4,3.5), we mentioned how the Cayley rotation parameterization cannot represent 180 degree rotations. Here we analyze other possible degenerate configurations for the solvers.

When d, the depth of the point in the reference image, is 0, the affine matrix A (Eq. (1)) goes to 0. When the normal vector is orthogonal to the vector from the reference camera to the 3D point, $m = 0$ and thus A is undefined. However, both of these configurations are impossible in real data.

We tested the P2ORI and UP1SIFT solvers with zero and near-zero rotation and/or translation but did not find any stability issues, unlike the P1AC solver, which has some instability with near-zero rotation and/or translation, depending

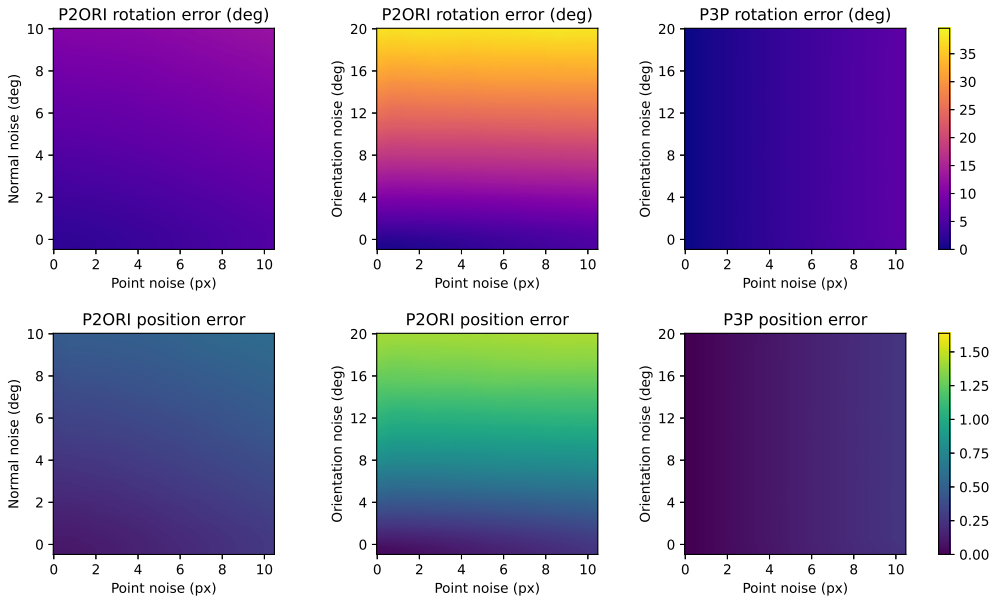

Figure 2. Median error of P2ORI and P3P solvers w.r.t. noise in the 2D point observations, normal vectors, and feature orientations.

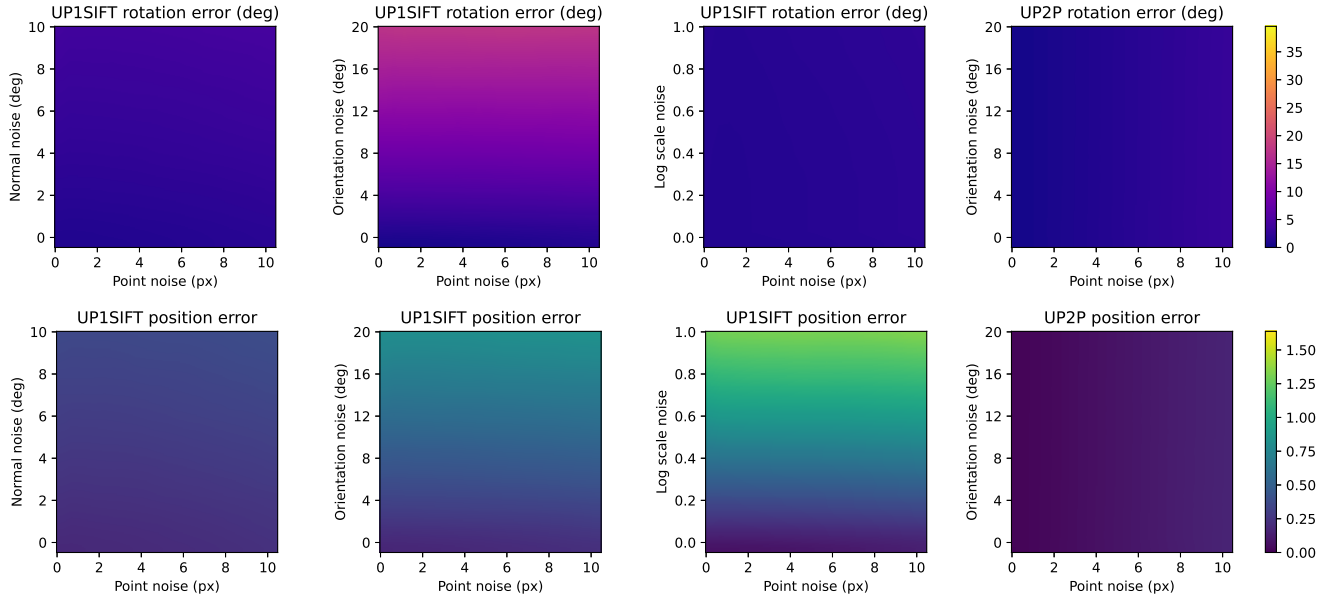

Figure 3. Median error of UP1SIFT and UP2P solvers w.r.t. noise in the 2D point observations, normal vectors, feature orientations, and feature scales.

on the 3Q3 implementation used [9].

4. Affine feature extraction

Affine feature extraction takes 1-2 seconds per image with AffNet [5] on a GPU. Other detectors, such as ASIFT [6], are even slower. One of the most important advantages of the proposed method compared to P1AC is that we do not need expensive affine shapes. We only need orientation and scale, which are obtained by default for many features. Estimating them, e.g., for learned detectors, is still more efficient than estimating affine shapes. Since scale and orientation provide an approximation to the full affine transformation, we decided to evaluate the P1AC solver on these approximate data rather than not comparing it at all.

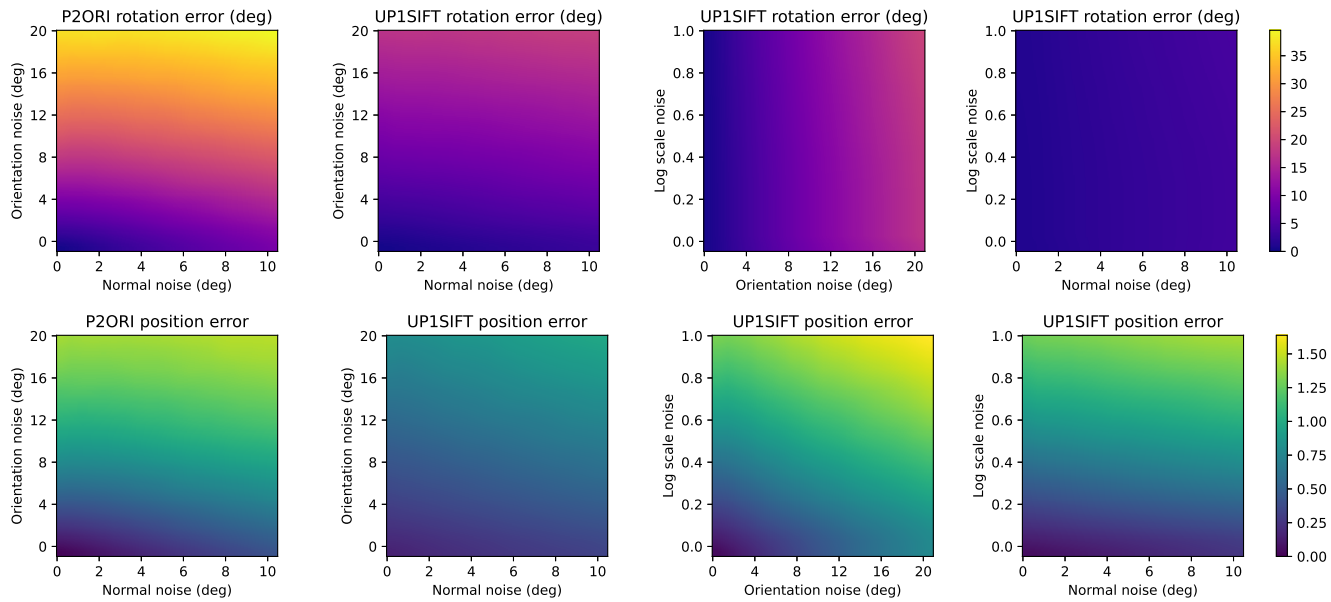

Figure 4. Median error of P2ORI and UPSIFT solvers w.r.t. increasing noise in the normal vectors, feature scales, and feature orientations.

5. Inlier scoring and pose refinement

For inlier scoring and pose refinement, we only used the point re-projection error and did not use the scale and orientation measurements. The scale and orientation tend to be noisy, and we have not found that using them for inlier scoring would improve the results. We left the investigation of their use for pose refinement as future work.

References

[1] Daniel Barath and Jiri Matas. Graph-Cut RANSAC: Local optimization on spatially coherent structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 4961–4974, 2021. 2

[2] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *Proceedings of the DAGM Symposium on Pattern Recognition*, pages 236–243, 2003. 2

[3] 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(6):381–395, 1981. 2

[4] Karel Lebeda, Jiri Matas, and Ondrej Chum. Fixing the locally optimized RANSAC. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2012. 2

[5] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision*, pages 284–300, 2018. 4

[6] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009. 4

[7] Torsten Sattler et al. RansacLib - a template-based *SAC implementation. https://github.com/tsattler/RansacLib, 2019. 2

[8] Phil H. S. Torr and Andrew Zisserman. Robust computation and parametrization of multiple view relations. In *International Conference on Computer Vision*, 1998. 2

[9] Jonathan Ventura, Zuzana Kukelova, Torsten Sattler, and Dániel Baráth. P1AC: Revisiting absolute pose from a single affine correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19751–19761, 2023. 4