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] Department of Computer Science, Computer Vision and Geometry Group, ETH Zürich

Abstract

Keypoints used for image matching often include an estimate of the feature scale and orientation. While recent work has demonstrated the advantages of using feature scales and orientations for relative pose estimation, relatively little work has considered their use for absolute pose estimation. We introduce minimal solutions for absolute pose from two oriented feature correspondences in the general case, or one scaled and oriented correspondence given a known vertical direction. Nowadays, assuming a known direction is not particularly restrictive as modern consumer devices, such as smartphones or drones, are equipped with Inertial Measurement Units (IMU) that provide the gravity direction by default. Compared to traditional absolute pose methods requiring three point correspondences, our solvers need a smaller minimal sample, reducing the cost and complexity of robust estimation. Evaluations on large-scale and public real datasets demonstrate the advantage of our methods for fast and accurate localization in challenging conditions. Code is available at `https://github.com/danini/absolute-pose-from-oriented-and-scaled-features`.

1. Introduction

The goal of absolute pose estimation is to determine the six degrees-of-freedom (6DOF) pose of an image from visual measurements. Absolute pose estimation is a core problem in computer vision with many applications, *e.g.*, visual localization [51], object recognition [18], Structure-from-Motion [53] and SLAM [11], and augmented reality [56].

The typical approach to absolute pose estimation is to obtain 2D-3D point correspondences by matching keypoints in a query image to 3D points in a scene and then computing the query image pose using a perspective-three-point (P3P) algorithm [15] inside a robust optimization loop, such as random sample consensus (RANSAC) [17] or more modern variants [6, 13]. The number of random samples required for RANSAC grows with both the outlier ratio and the minimal sample size exponentially. Thus, especially in difficult image matching situations where the outlier ratio is high, reducing the minimal sample size can lead to improvements in pose estimation speed and accuracy.

The P3P method relies on having at least three 2D observations in the query image of 3D points in the scene. Since the 3D scene points are typically obtained through triangulation, the 3D points are usually associated with 2D points in the reference images. Therefore, it is typical to have 2D-2D correspondences between the query image and reference images in addition to 2D-3D correspondences between the query image and the point cloud [48, 49]. These 2D-2D matches provide extra information for the absolute pose problem, which we exploit in our work to reduce the sample size needed for the pose estimation.

Previous work has examined how to reduce the minimal sample size for absolute pose by various means, such as introducing external pose information from an Inertial Measurement Unit (IMU) [25, 55], or leveraging extra information about the point correspondences such as local affine frames [24]. In particular, Ventura *et al.* showed that it is possible to compute the absolute pose from a single affine correspondence and an estimate of the surface normal at the 3D point [57]. However, affine covariant features are less commonly used as they are comparably expensive to compute, whereas the most widely-used feature detectors produce scale and orientation estimates [34, 47].

In this paper, we focus on designing absolute pose solvers that leverage scaled and oriented features, such as SIFT [33], to reduce the sample size. This is a highly practical scenario as most of the popular feature detectors output more than just the point locations by *default*. Our method is based on the equations connecting absolute pose to affine correspondences derived by Ventura *et al.* [57], and those relating affine features to scaled and oriented ones as established by Barath and Kukelova [5].

We derive novel constraints on the absolute pose from scaled and oriented features. Leveraging the proposed constraints, we design efficient minimal solutions for absolute pose, requiring two oriented features in the general case, or a single scale-and-orientation feature when the gravity direction is known, together with an estimate of the surface normal. The gravity direction can be measured using an IMU or from a vertical vanishing point [46], and thus is commonly available in recent consumer devices, *e.g*., when localizing robots, smartphones, and virtual reality headsets. Our contributions are as follows:

- We develop novel constraints on the absolute pose from scale-and-orientation features;
- We introduce two new minimal solvers for pose estimation from one or two scale-and-orientation features; and
- Through synthetic and real data experiments, we demonstrate that the proposed solvers lead to improved accuracy and run-time due to reducing the problem complexity when inserted into state-of-the-art robust estimators.

2. Related Work

The problem of estimating the absolute pose of a calibrated camera from three 2D-3D point correspondences has been considered for almost two centuries, as documented in a review of P3P solutions by Haralick *et al*. [19]. Recent work has continued to improve the speed and numerical stability of P3P solutions [21, 42], including a new state-of-the-art method introduced just recently [15].

The P3P approach uses only 2D point observations of 3D points. However, measurements of the local image transformation of the keypoints between the query image and the 3D surface, or between the query image and reference images, can also inform the absolute pose estimate. For example, Lowe [33] aggregated differences in position, scale, and orientation from SIFT [34] feature matches in a Hough transform to estimate the affine transformation of an object, and similar voting strategies have been used for outlier filtering [12] and spatial verification in image retrieval [54].

An affine correspondence (AC) includes an estimate of the affine transformation between the local image patches centred on the corresponding keypoint locations. Such correspondences can be estimated using affine shape adaptation [8], affine covariant region detectors [35, 37], or deep learning [39]. AC is equivalent to a first-order approximation of the local homography induced by the plane tangent to the surface at the 3D point [23, 24]. An affine correspondence provides three constraints on the fundamental matrix [9] or essential matrix [43] relating the reference and query images. Based on these and related findings, researchers have developed minimal solvers for relative pose using fewer correspondences than traditional point-based methods, such as estimating a homography [3] or the essential matrix [43] from two affine correspondences. Similarly, Köser and Koch introduced a solution for absolute pose estimation from a single affine correspondence from an orthorectified reference image [24], and Ventura *et al*. developed a general absolute pose solution from a perspective reference given knowledge of the surface normal [57].

However, most computer vision systems do not use affine covariant features, but instead rely on scale and orientation-covariant ones since they are faster to compute. Keypoint orientation can be estimated in various ways such as a histogram of gradients [34], the intensity centroid [44, 47], or supervised [61] or unsupervised learning [28]. Keypoint scale is often inherent to the keypoint detection process as in the Difference-of-Gaussian (DOG) detector employed by SIFT [34] or the Harris-Laplace corner detector [36], or can be estimated as part of the feature learning process [40, 60] or with a separately learned network [28].

Similar to affine covariant features, correspondences between scaled and oriented features provide constraints on the relative pose beyond the point correspondences. Such constraints can be used to reduce the sample size in minimal solvers. For example, Mills [38] described four-point solvers for essential matrix estimation, and Barath [2] developed a five-point solver for fundamental matrix estimation from oriented features. Barath and Kukelova [4, 5] rigorously defined the relationship between affine correspondences and scaled and oriented feature matches to produce a collection of constraints involving the affine transformation matrix and feature scales and orientations. They further used these constraints to design a two-point solver for homography estimation [4] and three- and four-point solvers for the essential and fundamental matrices, respectively [5].

To the best of our knowledge, no previous work has presented a minimal solution for absolute pose from scaled and oriented features. In this work, inspired by the affine decomposition introduced in [5] and the constraints on absolute pose from an AC proposed in [57], we develop a novel minimal absolute pose solver from two oriented points. We also introduce a novel solver for absolute pose from one scaled and oriented point, assuming a known gravity direction, which reduces the rotational unknowns to a single angle around the gravity axis. Previous work has developed two-point solvers for gravity-aware absolute pose [25, 55] and demonstrated their usefulness for image-based localization on smartphones and other IMU-equipped devices. The advantage of our solvers over the three-point and two-point solvers is that we require fewer points, thus reducing the sampling requirement in robust estimation.

3. Methods

Notation. We use a sans-serif capital letter M for a matrix, and an italic lower-case letter s for a scalar. We use subscripts to indicate matrix and vector indexing; e.g., $\mathsf{R}_{1:2,1:2}$ means the 2×2 upper-left submatrix of matrix R.

Let us consider a query camera, whose pose we want to estimate w.r.t. the world coordinate system. We assume we have a collection of reference images registered in the world coordinate system. Let $\mathsf{R}_{\text{query}}, \mathbf{t}_{\text{query}}$ be the unknown rotation and translation defining the world-to-camera transformation of the query image, and $\mathsf{R}_{\text{ref}}, \mathbf{t}_{\text{ref}}$ be the known rotation and translation of a reference image. The relative pose transformation from the reference image to the query image is $\mathsf{R} = \mathsf{R}_{\text{query}}\mathsf{R}_{\text{ref}}^T$ and $\mathbf{t} = \mathbf{t}_{\text{query}} - \mathsf{R}\mathbf{t}_{\text{ref}}$. We assume that all cameras under consideration are calibrated and thus do not include the intrinsics matrix in our formulations.

Our aim is to estimate $\mathsf{R}_{\text{query}}$ and $\mathbf{t}_{\text{query}}$. We assume we have established one or more feature matches between the query and reference images. Each feature match provides corresponding 2D points $\mathbf{p}_{\text{ref}}$ and $\mathbf{p}_{\text{query}}$ in the reference and query images, respectively.

In the case of having affine correspondences, for each correspondence we also have an estimate of the 2×2 local affine transformation matrix A which relates the local neighborhoods of the points such that $\mathbf{p}'_{\text{query}} = \mathbf{p}_{\text{query}} + \mathsf{A}(\mathbf{p}'_{\text{ref}} - \mathbf{p}_{\text{ref}})$, where $\mathbf{p}'_{\text{ref}}, \mathbf{p}'_{\text{query}}$ are points in the local neighborhoods of $\mathbf{p}_{\text{ref}}$ and $\mathbf{p}_{\text{query}}$, respectively. Note that affine correspondences are not required for our solver methods.

In the case of scaled and oriented features, for each correspondence, we have orientation angles $\alpha_{\text{ref}}, \alpha_{\text{query}}$ and scales $q_{\text{ref}}, q_{\text{query}}$ for the features centered on $\mathbf{p}_{\text{ref}}$ and $\mathbf{p}_{\text{query}}$ in the reference and query images, respectively.

Similar to Ventura et al. [57], we assume that we know the depth d of the point $\mathbf{p}_{\text{ref}}$ in the reference image and the normal vector $\mathbf{n}$ tangent to the surface at the 3D point. The depth could be obtained, for example, through triangulation, and the normal vector from the sparse point cloud or using deep learning methods [10].

3.1. Constraints from an affine correspondence on the absolute pose

Ventura *et al.* [57] derived the relationship between a local affine transformation and the pose of the query image relative to the reference image, *i.e.*, the relationship between A and $\mathsf{R}, \mathbf{t}$. The constraint is as follows:

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

where d is the depth of the point $\mathbf{p}_{\text{ref}}$ in the reference image, $\tilde{\mathbf{p}}_{\text{ref}} = [\mathbf{p}_{\text{ref}}^T \; 1]^T$, $\mathbf{n}_{\text{ref}} = \mathsf{R}_{\text{ref}}\mathbf{n}$ is the normal vector transformed to the coordinate system of the reference frame and $m = \mathbf{n}_{\text{ref}}^T\tilde{\mathbf{p}}_{\text{ref}}(d(\mathsf{R}_{3,:}\tilde{\mathbf{p}}_{\text{ref}}) + t_3)$.

3.2. Relationship between scaled and oriented features and affine correspondences

Barath and Kukelova [5] established several relationships between the affine transformation and the feature scales and orientations. The first relationship relevant to our approach arises from the transformation between the oriented circles centered on the point correspondence:

$$a_1 c_{\text{ref}} + a_2 s_{\text{ref}} - q c_{\text{query}} = 0, \tag{2}$$

$$a_3 c_{\text{ref}} + a_4 s_{\text{ref}} - q s_{\text{query}} = 0, \tag{3}$$

where $c_{\text{ref}} = \cos(\alpha_{\text{ref}}), s_{\text{ref}} = \sin(\alpha_{\text{ref}})$ and similarly for α_{query}, and $q = q_{\text{query}}/q_{\text{ref}}$. Eliminating q from Eqs. (2) and (3) yields a third constraint [4]:

$$c_{\text{ref}} s_{\text{query}} a_1 + s_{\text{ref}} s_{\text{query}} a_2 - c_{\text{ref}} c_{\text{query}} a_3 - c_{\text{query}} s_{\text{ref}} a_4 = 0. \tag{4}$$

The advantage of Eq. (4) over Eqs. (2) and (3) is that it only involves the feature orientations, not the scales, and thus can be applied when scales are unavailable, *e.g.*, when using oriented corner features such as ORB [47].

Note that since Eq. (4) is derived from Eqs. (2) and (3), they cannot be used together in a minimal solver, since satisfying Eqs. (2) and (3) also satisfies Eq. (4).

A second relevant relationship arises from the scale factor of the affine transformation [5]: $\det \mathsf{A} = q^2$. Expanding out the determinant gives a fourth constraint:

$$a_2 a_3 - a_1 a_4 + q^2 = 0 \enspace . \tag{5}$$

3.3. Constraints from scaled and oriented features on the absolute pose

Plugging the expression for the affine transformation from Eq. (1) into the scale and orientation constraints (Eqs. (2) to (5)) yields constraints on the absolute pose of the query camera involving only the feature orientations and scales.

We also have the two constraints arising from the point projection into the images as follows:

$$\mathbf{p}_{\text{query}_1}(d\mathsf{R}_{3,:}\tilde{\mathbf{p}}_{\text{ref}} + t_3) - (d\mathsf{R}_{1,:}\tilde{\mathbf{p}}_{\text{ref}} + t_1) = 0 \enspace , \tag{6}$$

$$\mathbf{p}_{\text{query}_2}(d\mathsf{R}_{3,:}\tilde{\mathbf{p}}_{\text{ref}} + t_3) - (d\mathsf{R}_{2,:}\tilde{\mathbf{p}}_{\text{ref}} + t_2) = 0 \enspace . \tag{7}$$

We explored various combinations of these six constraints and solution paths for the resulting systems of equations to find efficient and accurate minimal solvers for absolute pose. Next, we describe the two most practical and efficient solvers among those we found.

In contrast to the recent P1AC solver [57], in which the query camera pose was specified relative to the reference camera, we derive our solvers such that the query camera pose is in the world coordinate system. This is critical in a two-point solver since it allows for the two correspondences to come from different reference images, thus expanding the available samples for RANSAC.

3.4. Absolute pose from two oriented features

For the six DOF of the general absolute pose problem, we use two correspondences with three constraints each:

Eqs. (4), (6) and (7). The supplementary material (SM) contains a more complete discussion of possible combinations of constraints.

Note that Eqs. (6) and (7) are linear in the unknown rotation and translation parameters. To make Eq. (4) linear in the unknown parameters as well, we multiply both sides by m (from Eq. (1)). The SM provides a complete derivation of the system of equations.

Since the system of equations is linear in $\mathbf{t}_{\text{query}}$, we can use three equations to eliminate the translation parameters from the remaining three equations, *e.g.*, using Gauss-Jordan elimination. The remaining three equations will then be linear in the elements of $\mathsf{R}_{\text{query}}$ and will not contain $\mathbf{t}_{\text{query}}$.

To solve for the query rotation from these three equations, we apply the Cayley rotation parameterization with parameters x, y, z as follows:

$$\mathsf{R}_{\text{query}} = \frac{1}{s}\begin{bmatrix} 1+x^2-y^2-z^2 & 2(xy-z) & 2(y+xz) \\ 2(xy+z) & 1-x^2+y^2-z^2 & 2(yz-x) \\ 2(xz-y) & 2(x+yz) & 1-x^2-y^2+z^2 \end{bmatrix}, \tag{8}$$

where $s = 1 + x^2 + y^2 + z^2$. The Cayley parameterization is unable to represent 180° rotations. However, this degeneracy can be avoided by applying a random rotation to the variables [20, 27]. After multiplying the equations by s, we arrive at a system of three quadratic equations in x, y, z. Such a system is advantageous since an extremely efficient solver that significantly outperforms even efficient algebraic Gröbner basis or resultant-based solvers exists [26, 27, 63]. After finding up to eight solutions for $\mathsf{R}_{\text{query}}$ and retaining only real-valued ones, we find corresponding solutions for $\mathbf{t}_{\text{query}}$ through back-substitution.

3.5. Gravity-aware absolute pose from a scaled and oriented feature

In the gravity-aware case, we decompose the query camera rotation as $\mathsf{R}_{\text{query}} = \mathsf{R}_Y \mathsf{R}_{XZ}$, where the known rotation R_{XZ} aligns the Y axis of the world coordinate system with the observation of gravity in the camera's coordinate system, and R_Y is the remaining unknown rotation around the gravity direction. For the four DOF of the gravity-aware absolute pose problem, we use four constraints from a single observation: Eqs. (2), (3), (6) and (7).

Let θ be the unknown angle of rotation around the Y axis of the query camera, so that

$$\mathsf{R}_Y = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}. \tag{9}$$

We represent the angle using the tangent half-angle parameterization $r = \tan(\theta/2)$ which provides the substitutions $\cos(\theta) = \frac{1-r^2}{1+r^2}$ and $\sin(\theta) = \frac{2r}{1+r^2}$ [25]. After eliminating the translation variables, we arrive at a single quadratic equation in r, which is easily solved, giving two solutions for R_Y. We find corresponding solutions for $\mathbf{t}_{\text{query}}$ through back-substitution. Like the Cayley parameterization, the tangent half-angle parameterization cannot represent a 180° rotation. However, this can be similarly addressed by applying a random rotation to the variables.

3.6. Decomposition of an affine correspondence

In our synthetic data experiments, the situation arises where we have computed the affine transformation A for a randomly generated correspondence, and we wish to obtain feature scale and orientation values that are compatible with A. While a previous solution used random arbitrary values for some of the scale and orientation parameters [5], here we present a more principled approach via a minimal solver which finds multiple possible solutions.

Given an affine transformation A we want to decompose the transformation into two angles $\alpha_{\text{ref}}, \alpha_{\text{query}}$ and scales $q_{\text{ref}}, q_{\text{query}}$ such that Eqs. (2) to (5) are satisfied. Eq. (5) is satisfied by setting $q_{\text{ref}} = 1$ and $q_{\text{query}} = \sqrt{\det A}$. Note that we must have $\det A > 0$. Since Eq. (4) is derived from Eqs. (2) and (3), it will be satisfied when Eqs. (2) and (3) are satisfied. Therefore what remains is to find solutions for $\alpha_{\text{ref}}, \alpha_{\text{query}}$ that satisfy Eqs. (2) and (3).

We apply the tangent half-angle substitutions $r_{\text{ref}} = \tan\frac{\alpha_{\text{ref}}}{2}$ and $r_{\text{query}} = \tan\frac{\alpha_{\text{query}}}{2}$. Eqs. (2) and (3) become

$$a_1 \frac{1-r_{\text{ref}}^2}{1+r_{\text{ref}}^2} + a_2 \frac{2r_{\text{ref}}}{1+r_{\text{ref}}^2} - q\frac{1-r_{\text{query}}^2}{1+r_{\text{query}}^2} = 0 \ , \tag{10}$$

$$a_3 \frac{1-r_{\text{ref}}^2}{1+r_{\text{ref}}^2} + a_4 \frac{2r_{\text{ref}}}{1+r_{\text{ref}}} - q\frac{2r_{\text{query}}}{1+r_{\text{query}}^2} = 0 \ . \tag{11}$$

After multiplying terms to eliminate the denominators, we end up with two polynomials in $r_{\text{ref}}, r_{\text{query}}$ of degree four. We used the GAPS package [31] to automatically produce a solver for the system. The solver produces eight possible solutions for $r_{\text{ref}}, r_{\text{query}}$ from which we remove any solutions containing an imaginary part. Since, for the purposes of our synthetic data experiments, any solution is acceptable, we simply use the first real-valued solution returned.

4. Experiments

In our experiments, we compare the following methods:

- P2ORI (Sec. 3.4): Our novel absolute pose solver from two oriented feature correspondences.
- UP1SIFT (Sec. 3.5): Our novel gravity-aware absolute solver from one scaled and oriented feature match[1].
- P1AC [57]: A solver for absolute pose from a single affine correspondence.

[1] We reference SIFT [34] in the name since it is arguably the most famous scaled and orientation feature, but UP1SIFT can be used with any feature providing a scale and orientation estimate.

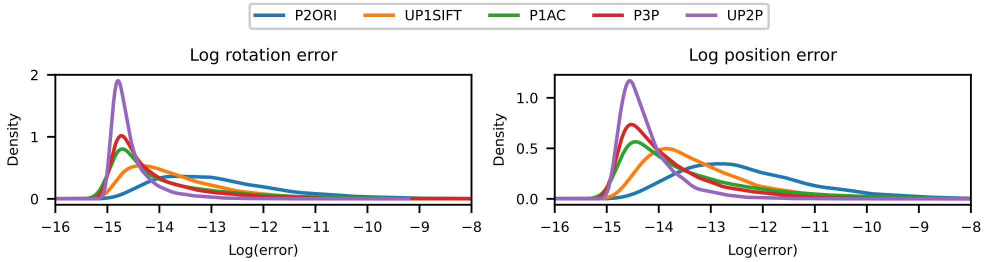

Figure 1. Results of numerical stability experiment, showing density of log rotation and position errors with zero noise added to the observations. All solvers are numerically stable.

- P3P [15]: A solver for absolute pose from three point correspondences.
- UP2P [25]: A solver for gravity-aware absolute pose from two point correspondences.

We evaluate each solution using two standard metrics: rotation error and position error [52, 57]. Given the true query rotation $\mathsf{R}_{\text{query}}$ and the estimated rotation $\tilde{\mathsf{R}}_{\text{query}}$, the rotation error is computed as $||\ln(\tilde{\mathsf{R}}_{\text{query}}\mathsf{R}^T_{\text{query}})||$. Given the true query camera center $\mathbf{c}_{\text{query}} = -\mathsf{R}^T_{\text{query}}\mathbf{t}_{\text{query}}$ and the estimated query camera center $\tilde{\mathbf{c}}_{\text{query}}$, the position error is calculated as $||\mathbf{c}_{\text{query}} - \tilde{\mathbf{c}}_{\text{query}}||$. In the synthetic data experiments, when a method returns multiple solutions, we choose the solution that minimizes the maximum of these two errors.

4.1. Synthetic data

To evaluate our solvers in terms of numerical stability and sensitivity to noise, we tested them on randomly generated synthetic problem instances.

To generate each synthetic data problem, we used the following setup [16, 57]. We select camera centers for the reference and query cameras at random positions at a distance randomly chosen between 1 and 2 from the origin. We choose a random target point within $[-0.5\ 0.5]^3$ and orient each camera to look at the target point. To generate a point correspondence between the cameras, we select a random 3D point from $\mathcal{N}(\mathbf{0}, \mathsf{I}_{3\times3})$ and project it to the two cameras. We select a random normal vector for the point and use it to calculate the local homography induced by the plane tangent the surface at the point. We then extract the affine transformation matrix from the local homography [3], and use our decomposition method (Sec. 3.6) to extract scales and orientations from the affine transformation.

We discard any problem configurations where: the rotation between the reference and query cameras is greater than 180 degrees; the 3D point is behind either camera; or the determinant of the affine transformation is not positive. Such scenarios do not appear in real world experiments.

To support the application of the gravity-aware solvers we decompose the query rotation $\mathsf{R}_{\text{query}}$ into a rotation around the Y-axis R_Y and a rotation R_{XZ} around a vector in the X-Z plane such that $\mathsf{R}_{\text{query}} = \mathsf{R}_Y\mathsf{R}_{XZ}$. R_{XZ} is provided as an input to the solvers and the only rotational unknown is θ, the amount of rotation around the Y-axis.

P2ORI	P1AC [57]	P3P [42]	UP1SIFT	UP2P [26]
2.62	1.92	0.42	1.05	0.23

Table 1. Average timing in μs over 10,000 trials.

Numerical Stability. We tested each method on 10,000 random problem configurations with zero noise added to the observations to test the numerical stability of each solver. Fig. 1 shows a density plot of the log rotation and position error of each solver. Note that the aim of this experiment is not to compare the accuracy of the solvers but only to establish that each is numerically stable with no peaks above a reasonable level, such as 1×10^{-5}. As shown in the figure, the median errors of all solvers are below 1×10^{-12}.

Timings. We measured the average time to solve a single problem configuration for each solver over 10,000 random problems. The measurements are given in Tab. 1. Each solver was implemented in C++ and the timings were measured on an Apple M1 Pro MacBook with 16 GB RAM. The implementations of P3P and UP2P came from PoseLib [27] and the implementation of P1AC from the authors' official code release. Among general absolute pose solvers, P2ORI is slower than P1AC and P3P; UP1SIFT is more than twice as fast as P2ORI but still slower than UP2P. However, all solvers are very fast (under 3 μs). Furthermore, in robust optimization the other steps such as inlier counting and local optimization are far more computationally costly, *i.e.*, the measured differences hardly impact the overall run-times.

Noise Experiments.

We tested the sensitivity of each solver to various types of observation noise:

- *Point noise*: Gaussian noise added to the 2D point observations with a focal length of 400 pixels.
- *Normal noise*: a random rotation with normally distributed angle applied to the normal vector $\mathbf{n}$.
- *Orientation noise*: Gaussian noise added to α_{query}.

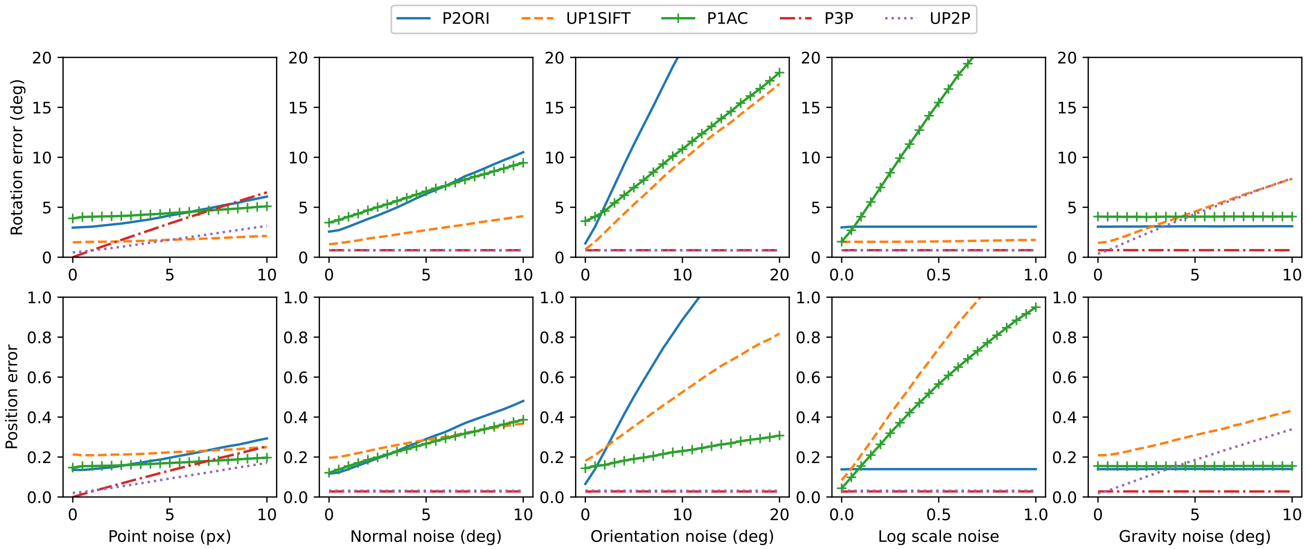

Figure 2. Median error w.r.t. noise in the 2D point observations, normal vectors, feature orientations, feature scales, and gravity vector. The x-axis shows the std. dev. of the noise added.

- *Log scale noise*: Gaussian noise added to $\log q$.
- *Gravity noise*: a random rotation with normally distributed angle applied to R_{XZ}.

We tested each solver across a range of noise settings for each noise type, computing the median error over 10,000 iterations at each setting. For the noise types not being varied, we applied a default noise setting based on the level of noise expected in real-world settings: 1 pixel point noise; 1 deg normal noise; 1 degrees orientation noise; 0.1 log scale noise; and 0.5 deg gravity noise. We tested a wide range of noise values since the actual noise in the observations depends on many factors including the camera configuration, feature detection algorithm, and IMU sensor characteristics. For example, the measurement noise of the gravity direction varies from $0.5°$ in a low-cost IMU to $0.02°$ in a high accuracy IMU [25]. Barath *et al.* [7] estimated the orientation noise of SIFT [34] features to be $11.8°$ and the log scale noise to be 0.51, based on analysis of a large-scale homography benchmark dataset; however, modern learned features have substantially more accurate scale and orientation estimates [28].

The results for varying one noise type at a time are shown in Fig. 2. In general, we can conclude that P2ORI is most sensitive to orientation noise, while UP1SIFT's position estimate is most sensitive to orientation and log scale noise, and UP1SIFT's rotation estimate is most sensitive to log scale noise. The SM includes analyses varying two noise types simultaneously.

4.2. Real data experiments

Datasets. To evaluate the performance of the compared methods in large-scale image-based localization, we utilized the Cambridge Landmarks [22] and Aachen Day-Night v1.1 [50, 52, 62] benchmark datasets, both well-regarded in the literature of visual localization.

The Cambridge Landmarks dataset consists of six scenes in Cambridge, UK, each recorded via multiple video sequences taken with a smartphone, capturing different parts of the city. From the recorded sequences for each scene, some provide database images representing the scene, while others are used to acquire query images. Ground truth poses and intrinsic camera calibrations for all images were determined using the VisualSFM [58, 59] Structure-from-Motion (SfM) software. Localization performance is commonly evaluated by reporting the median position and orientation error. Additionally, we assess the proportion of images (*i.e.*, recall) localized within 5cm / 1°, 10cm / 1°, and 20cm / 1° of their actual poses.

Cambridge Landmarks is generally less challenging because each scene is small, and query images were captured around the same time as the database images. One scene, Street, is an exception to this, posing a challenge to most feature matchers and absolute pose estimators.

The Aachen Day-Night dataset, representing the historic city center of Aachen (Germany), is more challenging than Cambridge Landmarks. Besides its larger scale, it includes daytime database images and nighttime query images taken over an extended period of time. The ground truth for the daytime images was established via COLMAP [53], with nighttime queries aligned by refining initial pose es-

Scene	Position (cm) ↓				Rotation (°) ↓				Recall (0.05m/1°) ↑				Recall (0.1m/1°) ↑			
	P3P	P1AC	P2ORI	UP1SIFT	P3P	P1AC	P2ORI	UP1SIFT	P3P	P1AC	P2ORI	UP1SIFT	P3P	P1AC	P2ORI	UP1SIFT
Great Court	**2**	**2**	**2**	**2**	**0.01**	**0.01**	**0.01**	**0.01**	90.4	92.0	**93.0**	91.7	96.5	**97.2**	96.5	96.2
King's Col.	**1**	2	**1**	2	**0.02**	0.03	0.03	0.03	89.2	88.1	**89.5**	86.0	97.1	**98.0**	97.7	96.2
Old Hospital	**2**	3	**2**	3	**0.04**	**0.04**	**0.04**	0.05	80.2	**82.4**	81.3	73.6	95.1	**97.3**	96.2	93.4
Shop Façade	**1**	**1**	**1**	**1**	**0.06**	**0.06**	**0.06**	**0.06**	91.3	87.9	**96.1**	88.4	99.0	99.1	**100.0**	99.0
St Mary's Ch.	**2**	**2**	**2**	**2**	0.06	**0.05**	**0.05**	**0.05**	84.7	87.9	**90.4**	87.7	94.2	95.1	**97.4**	96.2
Street	197	40	21	**6**	2.57	1.12	0.66	**0.20**	7.8	19.0	30.0	**46.3**	18.2	26.8	38.3	**52.7**
Avg.	34	8	5	**3**	0.46	0.23	**0.14**	0.17	73.9	76.5	**80.1**	79.0	83.3	85.7	87.7	**88.8**
Weighted avg.	120	25	14	**4**	1.56	0.74	0.41	**0.13**	39.5	46.8	54.1	**62.7**	49.0	54.6	61.6	**69.9**

Table 2. **Cambridge Landmarks** [22] median position (centimeters) and rotation (degrees) errors, and recalls (percentages) at 0.05m/1° and 0.1m/1°, of GC-RANSAC [6] combined with various solvers when using SuperPoint [14] + LightGlue [32] + SelfScaleOri [28] matches. The average over all scenes and average weighted by the number of images in each scene are in the two last rows.

	Recall (0.05m/1°) ↑					Recall (0.1m/1°) ↑					Recall (0.2m/1°) ↑				
	P3P	UP2P	P1AC	P2ORI	UP1SIFT	P3P	UP2P	P1AC	P2ORI	UP1SIFT	P3P	UP2P	P1AC	P2ORI	UP1SIFT
ORB-2k	21.8	20.2	30.0	35.4	**40.7**	37.4	36.1	48.1	53.1	**57.5**	47.7	49.5	62.2	65.6	**68.2**
RootSIFT-2k	56.3	45.6	56.8	57.0	**58.5**	66.9	56.0	71.9	70.1	**73.5**	72.4	60.4	79.6	77.0	**81.0**
RootSIFT-8k	64.3	64.2	65.9	70.6	**71.7**	76.1	73.0	79.1	82.3	**82.9**	81.5	76.5	86.3	87.2	**88.6**
SP + LG + SSO	73.9	76.2	76.5	**80.1**	79.0	83.3	86.4	85.7	87.7	**88.8**	87.0	90.5	88.6	90.6	**92.6**

Table 3. **Cambridge Landmarks** [22] recalls (in percentages) at 0.05m/1°, 0.1m/1° and 0.2m/1°, of GC-RANSAC [6] combined with various solvers on ORB-2k [47], RootSIFT-2k and 8k [34], and SuperPoint [14] + LightGlue [32] + SelfScaleOri [28] matches. Bold numbers indicate the best performing approach and underlined numbers the second best.

Scene	Time (secs) ↓				
	P1AC	P3P	UP2P	P2ORI	UP1SIFT↓
Great Court	1.18	5.72	1.38	1.22	**0.68**
King's College	1.78	8.40	1.42	2.01	**0.99**
Old Hospital	1.07	2.95	3.18	1.02	**0.42**
Shop Facade	1.99	5.64	1.63	1.33	**1.29**
St Mary's Church	2.49	4.11	5.56	2.00	**0.60**
Street	2.30	0.80	1.71	2.16	**0.20**
Day	2.47	2.81	2.48	1.45	**1.34**
Night	1.29	3.42	1.75	2.15	**0.20**

Table 4. Average times of GC-RANSAC [6] on the Cambridge Landmarks [22] and Aachen Day-Night [50] datasets on SuperPoint [14] + LightGlue [32] + SelfScaleOri [28] matches.

timates [62]. Our evaluation adheres to the standard protocol, reporting the percentage of images localized within three error thresholds (0.25m / 2°, 0.5m / 5°, 5.0m / 10°).

Features. There are multiple ways of obtaining orientations and scales for image features from real images. One of the most representative methods is SIFT [34], which introduced gradient histograms for orientation estimation, while the scale estimates come from the image scale pyramid where a particular Difference-of-Gaussian (DoG) feature is found. Rublee et al. [47] proposed an efficient measure of corner orientation using intensity centroid on the FAST detectors [45]. Self-Scale-Ori [28] and its variants [29, 30] tackle the local feature orientation and scale estimation via a learning-based approach applicable to any feature detector as a post-processing step, applied on the detected keypoints.

In the experiments, we use DoG features [34] with RootSIFT [1] descriptors and the default estimated orientations and scales. We test two versions, one obtaining the 2k best matches and one with the 8k best ones. We also run ORB [47] with 2k features. Additionally, we detect SuperPoint features [14], establish matches by the recent LightGlue [32], and obtain scales and orientations by Self-Scale-Ori [28].

Normal vectors were estimated using 200 nearest neighbors for each point in the SFM point cloud.

Competitors. We compare the proposed P2ORI and UP1SIFT solvers with P3P [42] and UP2P [25], where both UP1SIFT and UP2P exploit the gravity direction for estimating the absolute pose. Moreover, we compare to the recent P1AC solver [57], which requires a single affine correspondence to estimate the pose.

As we do not have affine correspondences, only scales and orientations, to apply P1AC we approximate the affine frame A_i for the i-th feature as $\mathsf{A}_i = \mathsf{S}_{q_i}\mathsf{R}_{\alpha_i}$, where S_{q_i} is a diagonal matrix, scaling uniformly along the axes by the detected scale factor q_i, and $\mathsf{R}_{\alpha_i} \in \mathrm{SO}(2)$ rotates by the estimated angle $\alpha_i \in [0, 2\pi)$. See the SM for more details.

Robust estimation method. We estimate the query pose from correspondences from multiple reference images [41] using GC-RANSAC [6] for robust estimation. The SM includes details about inlier scoring and pose refinement.

Results.

The median position and rotation errors and recalls on the Cambridge Landmarks dataset, using Su-

	Day	Night
P3P	55.6 / 74.2 / 93.7	46.1 / 56.5 / 68.1
UP2P	54.0 / 70.8 / 87.6	19.4 / 29.3 / 47.1
P1AC	60.0 / 80.1 / 93.4	46.1 / 57.1 / 70.2
P2ORI	**60.9** / 81.2 / **95.0**	40.8 / 53.9 / 61.8
UP1SIFT	60.3 / **82.4** / 94.8	**47.6 / 58.1 / 71.2**

Table 5. **Aachen Day-Night** pose error recalls [50], in percentages, at 0.25m/2°, 0.5m/5°, and 5.0m/10° for GC-RANSAC [6] combined with various solvers on RootSIFT matches.

	Day	Night
P3P	60.6 / 79.6 / 92.4	59.7 / 77.5 / 92.7
UP2P	65.8 / 86.8 / 96.8	67.5 / 86.4 / 97.9
P1AC	64.4 / 83.1 / **97.5**	69.6 / 87.4 / **99.0**
P2ORI	**68.2 / 87.3** / 96.6	**71.7** / 86.4 / **99.0**
UP1SIFT	66.3 / 85.8 / 97.0	69.6 / **88.0 / 99.0**

Table 6. **Aachen Day-Night** pose error recalls [50] at 0.25m/2°, 0.5m/5°, and 5.0m/10° for GC-RANSAC [6] combined with various solvers on SuperPoint + LightGlue + SelfScaleOri matches.

perPoint+LightGlue+SelfScaleOri (SP+LG+SSO) features, are reported in Table 2. For this test, we use the ground truth gravity direction obtained from the reference absolute poses for UP2P and UP1SIFT. While the results on most scenes are similarly accurate for all methods, we can generally say that the proposed P2ORI leads to particularly good accuracy, on par with the UP1SIFT solver that uses the ground truth gravity direction. The differences are more pronounced in scene Street, where the two proposed solvers are the best by a considerable margin compared to P3P and P1AC. Interestingly, the P1AC solver works surprisingly well, even on approximated affine correspondences. We also report average and weighted (by the number of queries in a scene) average errors at the bottom of the table. The UP1SIFT solver achieves good results; however, it uses the ground truth gravity direction. The P2ORI method is the second best, being ahead of P3P and P1AC by a large margin. The average recalls of UP2P are 76.2 (0.05m/1°) and 86.4 (0.1m/1°), being less accurate than P2ORI and UP1SIFT.

We also report the processing times of the robust estimation in Table 1. As expected, the solvers requiring a single correspondence (P1AC and UP1SIFT) run the fastest. The P2ORI method is also significantly faster than using P3P.

The recall values when using different feature types are shown in Table 3. While it is clear that SP+LG+SSO performs the best, the proposed solvers improve upon P3P, UP2P, and P1AC, independently of the features employed.

Given that Aachen Day-Night is a more challenging dataset, we only test the solvers with RootSIFT-8k and SP+LG+SSO features. Since we do not know the gravity direction, we assume that it points downward (it is $[0, -1, 0]^T$). This usually is a safe assumption as humans tend to take pictures upright. The results of using RootSIFT-8k features are shown in Table 5. On the Day images, the proposed P2ORI, UP1SIFT, and P1AC achieve the highest recalls, with P2ORI being the best by a small margin. This clearly shows the importance of using small minimal samples in RANSAC. Despite the inaccurate gravity prior, UP1SIFT excels due to requiring only a single correspondence. The same holds for P1AC, which only gets an approximated affine correspondence and still obtains reasonably accurate results. This is even more pronounced in the Night images, where UP1SIFT is the most accurate method and P1AC is the second most accurate.

Results with SP+LG+SSO correspondences are in Table 6. Since SP+LG+SSO are strong features that solve the Day scenes with close to 100% recall, we resized the images so that their longest dimension is 800 pixel at maximum. On the Day sequence, all methods achieve similarly good accuracy, except for P3P, which lags slightly behind. The best performance is achieved by the proposed P2ORI method. While P1AC achieves high recall at 5.0m / 10°, it is inaccurate at the stricter thresholds. On the Night sequence, the proposed P2ORI and UP1SIFT achieve high accuracy.

Regarding the computational overhead of computing scale and orientation estimates for learned features, SuperPoint takes 0.39 seconds per image on average, and SelfScale-Ori takes 0.20 seconds in our experiments.

5. Conclusions

In this work, we explored various constraints on the calibrated absolute camera pose from scaled and oriented point correspondences, assuming knowledge of surface normals. Based on these constraints, we developed two novel minimal solvers: P2ORI for absolute pose from two oriented points, and UP1SIFT for gravity-aware absolute pose from one scaled and oriented point.

Like all solvers based on region correspondences rather than point correspondences, our solvers are susceptible to more types of noise than point-based solvers. However, when applied with robust estimation methods on standard localization benchmarks, our solvers produce more accurate pose estimates and higher recall than the standard point-based approaches (P3P and UP2P) and P1AC while also being on average as fast or faster.

Future work includes exploring solvers for uncalibrated, generalized, and non-minimal problems and integrating our methods into SfM and SLAM systems.

Acknowledgments. This work was supported by NSF Award No. 2144822, EU Horizon 2020 project RICAIP (No. 857306), and the Czech Science Foundation (GAČR) JUNIOR STAR Grant No. 22-23183M. D. Barath was supported by the Hasler Stiftung Research Grant via the ETH Zurich Foundation.

References

[1] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012. 7

[2] Daniel Barath. Five-point fundamental matrix estimation for uncalibrated cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 235–243, 2018. 2

[3] Daniel Barath and Levente Hajder. Novel ways to estimate homography from local affine transformations. In *Proceedings of the Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 434–445, 2016. 2, 5

[4] Daniel Barath and Zuzana Kukelova. Homography from two orientation-and scale-covariant features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1091–1099, 2019. 2, 3

[5] Daniel Barath and Zuzana Kukelova. Relative pose from SIFT features. In *Proceedings of the European Conference on Computer Vision*, pages 454–469. Springer, 2022. 1, 2, 3, 4

[6] 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. 1, 7, 8

[7] Daniel Barath, Dmytro Mishkin, Michal Polic, Wolfgang Förstner, and Jiri Matas. A large-scale homography benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21360–21370, 2023. 6

[8] Adam Baumberg. Reliable feature matching across widely separated views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 774–781, 2000. 2

[9] Jacob Bentolila and Joseph M Francos. Conic epipolar constraints from affine correspondences. *Computer Vision and Image Understanding*, 122:105–114, 2014. 2

[10] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 3

[11] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics*, 37(6): 1874–1890, 2021. 1

[12] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Handcrafted outlier detection revisited. In *European Conference on Computer Vision*, 2020. 2

[13] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *Proceedings of the DAGM Symposium on Pattern Recognition*, pages 236–243, 2003. 1

[14] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 7

[15] Yaqing Ding, Jian Yang, Viktor Larsson, Carl Olsson, and Kalle Åström. Revisiting the P3P problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4872–4880, 2023. 1, 2, 5

[16] Iván Eichhardt and Dmitry Chetverikov. Affine correspondences between central cameras for rapid relative pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 482–497, 2018. 5

[17] 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. 1

[18] Qiang Hao, Rui Cai, Zhiwei Li, Lei Zhang, Yanwei Pang, Feng Wu, and Yong Rui. Efficient 2D-to-3D correspondence filtering for scalable 3D object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 899–906, 2013. 1

[19] Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13:331–356, 1994. 2

[20] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (DLS) method for PnP. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 383–390, 2011. 4

[21] Tong Ke and Stergios I Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7225–7233, 2017. 2

[22] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015. 6, 7

[23] Kevin Köser. *Geometric estimation with local affine frames and free-form surfaces*. PhD thesis, University of Kiel, 2009. 2

[24] Kevin Köser and Reinhard Koch. Differential spatial resection-pose estimation using a single local image feature. In *Proceedings of the European Conference on Computer Vision*, pages 312–325, 2008. 1, 2

[25] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *Asian Conference on Computer Vision*, pages 216–229. Springer, 2010. 1, 2, 4, 5, 6, 7

[26] Zuzana Kukelova, Jan Heller, and Andrew Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1799–1808, 2016. 4, 5

[27] Viktor Larsson. PoseLib - Minimal Solvers for Camera Pose Estimation. `https://github.com/vlarsson/PoseLib`, 2020. 4, 5

[28] Jongmin Lee, Yoonwoo Jeong, and Minsu Cho. Self-supervised learning of image scale and orientation. In *Proceedings of the British Machine Vision Conference*, 2021. 2, 6, 7

[29] Jongmin Lee, Byungjin Kim, and Minsu Cho. Self-supervised equivariant learning for oriented keypoint detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4847–4857, 2022. 7

[30] Jongmin Lee, Byungjin Kim, Seungwook Kim, and Minsu Cho. Learning rotation-equivariant features for visual correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21887–21897, 2023. 7

[31] Bo Li and Viktor Larsson. GAPS: Generator for automatic polynomial solvers. *arXiv preprint arXiv:2004.11765*, 2020. 4

[32] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. *Proceedings of the International Conference on Computer Vision*, 2023. 7

[33] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1150–1157, 1999. 1, 2

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 2, 4, 6, 7

[35] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004. 2

[36] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 525–531, 2001. 2

[37] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60:63–86, 2004. 2

[38] Steven Mills. Four-and seven-point relative camera pose from oriented features. In *2018 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2018. 2

[39] 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. 2

[40] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. *Advances in neural information processing systems*, 31, 2018. 2

[41] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. MeshLoc: Mesh-based visual localization. In *European Conference on Computer Vision*, 2022. 7

[42] Mikael Persson and Klas Nordberg. Lambda Twist: An accurate fast robust perspective three point (P3P) solver. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–332, 2018. 2, 5, 7

[43] Carolina Raposo and Joao P Barreto. Theory and practice of structure-from-motion using affine correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5470–5478, 2016. 2

[44] Paul L Rosin. Measuring corner properties. *Computer Vision and Image Understanding*, 73(2):291–307, 1999. 2

[45] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision*, pages 430–443. Springer, 2006. 7

[46] Carsten Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10):647–655, 2002. 2

[47] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Ccomputer Vision*, pages 2564–2571, 2011. 1, 2, 3, 7

[48] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1

[49] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1

[50] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *Proceedings of the British Machine Vision Conference*, page 4, 2012. 6, 7, 8

[51] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2016. 1

[52] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF urban visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 5, 6

[53] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 1, 6

[54] Johannes L Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Proceedings of the Asian Conference on Computer Vision*, pages 321–337, 2017. 2

[55] Chris Sweeney, John Flynn, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. Efficient computation of absolute pose for gravity-aware augmented reality. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 19–24, 2015. 1, 2

[56] Jonathan Ventura and Tobias Höllerer. Wide-area scene mapping for mobile visual tracking. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12, 2012. 1

[57] 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. 1, 2, 3, 4, 5, 7

[58] Changchang Wu. Towards linear-time incremental structure from motion. In *Proceedings of the IEEE International Conference on 3D Vision*, pages 127–134, 2013. 6

[59] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz. Multicore bundle adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3057–3064, 2011. 6

[60] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision*, pages 467–483, 2016. 2

[61] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit. Learning to assign orientations to feature points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 107–116, 2016. 2

[62] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision*, 129:821–844, 2021. 6, 7

[63] Lipu Zhou, Jiamin Ye, and Michael Kaess. A stable algebraic camera pose estimation for minimal configurations of 2D/3D point and line correspondences. In *Proceedings of the Asian Conference on Computer Vision*, pages 273–288, 2018. 4