P1AC: Revisiting Absolute Pose From a Single Affine Correspondence

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

Abstract

Affine correspondences have traditionally been used to improve feature matching over wide baselines. While recent work has successfully used affine correspondences to solve various relative camera pose estimation problems, less attention has been given to their use in absolute pose estimation. We introduce the first general solution to the problem of estimating the pose of a calibrated camera given a single observation of an oriented point and an affine correspondence. The advantage of our approach (P1AC) is that it requires only a single correspondence, in comparison to the traditional point-based approach (P3P), significantly reducing the combinatorics in robust estimation. P1AC provides a general solution that removes restrictive assumptions made in prior work and is applicable to large-scale image-based localization. We propose a minimal solution to the P1AC problem and evaluate our novel solver on synthetic data, showing its numerical stability and performance under various types of noise. On standard image-based localization benchmarks we show that P1AC achieves more accurate results than the widely used P3P algorithm. Code for our method is available at https://github.com/jonathanventura/P1AC/.

1. Introduction

Image-based localization is the process of determining the pose of a query image in a reference coordinate system from pre-registered images in a database. Image-based localization is an important technology for many applications including robotics [41], self-driving cars [20] and augmented reality [51]. It also forms an essential component of large-scale structure-from-motion pipelines [48].

The dominant method for image-based localization is to estimate the camera pose from correspondences between 2D features in the query image and known 3D points in the scene. Since these correspondences are likely to contain mismatches, a best-fit model is found through Random Sample Consensus (RANSAC) [17] or one of its derivatives [12, 32, 5], where camera pose hypotheses are estimated from minimal samples of three point correspondences (PCs) using a perspective-three-point (P3P) algorithm [44].

In cases of very wide-baseline matching or with extreme differences in viewing angle, affine-covariant features increase match reliability in comparison to traditional difference-of-Gaussian detectors such as SIFT [35]. A match between affine-covariant features, which we will call an affine correspondence (AC), estimates a $2 \times 2$ affine transformation matrix between the local regions around the corresponding points in the two images.

Recently, several authors have developed a theory of the constraints induced by ACs on two-view geometry [9, 45, 16], and minimal solvers for relative pose estimation from ACs [16, 15]. However, relatively little attention has been given to absolute pose estimation from ACs. We revisit the problem of estimating absolute camera pose from a single affine correspondence, and refer to this
as the perspective-one-affine-correspondence (P1AC) problem. Köser and Koch [29] considered a limited form of the P1AC problem by assuming that the reference view is orthorectified. In contrast, our method removes this limiting assumption and makes more general assumptions: that the reference and query images are perspective images of a locally planar surface in an arbitrary orientation, that we have estimated the normal to the surface at the observed 3D point, and that we have estimated an affine transformation between the reference and the query images. We derive a system of equations for this problem and propose the first general solution for absolute pose from a single AC.

In comparison to P3P solutions, our P1AC solution only requires a single correspondence instead of three in the minimal case, and thus significantly reduces the complexity of robust estimation. We demonstrate these advantages through experiments on synthetic and real data.

Our contributions are as follows:
- We introduce a novel solution to the P1AC problem using reduction to the 3Q3 problem [30]. Our solution is general and does not impose the restrictive assumptions made in previous work.
- Experiments on synthetic data establish the numerical stability of our solver and analyze its performance under various types of noise.
- Experiments on two benchmark datasets for image-based localization show that our P1AC solution produces more accurate pose estimates than P3P.

2. Related work

Affine correspondences are commonly estimated using either frame-to-frame tracking techniques [36, 3] or affine-covariant region detection [37, 39]. In the case of frame-to-frame tracking, the deformation of a locally planar surface observed by a moving camera is well-modeled by an affine transformation. For affine-covariant regions, an affine transformation describing the shape of the local region around the interest point is estimated [8, 39]. While computing ACs may increase feature detection computation time over computing PCs, ACs improve match reliability in wide-baseline image matching [24], introduce new constraints on camera geometry [28, 9] and, therefore, decrease the number of correspondences needed for camera pose estimation [29, 45, 16, 15].

Recent work has examined how ACs might be used to improve and simplify pose estimation and subsequently improve image-based localization, visual odometry, and structure-from-motion. Köser developed a theory of relationship between homographies and ACs [28], and Bentolila and Francos demonstrated that an AC provides three constraints on the fundamental matrix [9]. Raposo and Barreto re-derived these constraints [45], making them usable for essential matrix estimation using only two ACs instead of five PCs as is normally needed [42]. Eichhardt and Chetverikov extended relative pose estimation using ACs to handle any central camera model [16]. Eichhardt and Barath [15] demonstrated relative pose estimation from a single AC and a depth map estimated by a neural network. Recently, Barath et al. provided recommendations for speeding up RANSAC when using ACs [6], including local feature geometry refinement and uncertainty propagation in RANSAC. The ideas presented in [6] can be used in combination with any pose estimation solver that uses ACs, including the solvers proposed in this paper.

Absolute pose estimation using ACs has received less attention than relative pose. Algorithms for absolute pose estimation from PCs have a long history, dating back to Grunert’s solution [18]. Among many recent P3P solutions [27, 25, 44, 14] the Lambda Twist method [44] and [14] are among the fastest and most numerically stable.

Köser and Koch introduced a solution for absolute pose from a single AC called Differential Pose Resection (DPR), with the restriction that the reference image is a scaled orthographic image of a planar surface with the normal parallel to the optical axis [29]. In our setting, the reference image is, instead, a perspective image of a locally planar surface in an arbitrary orientation. Their method could be applied in our setting if we rectified the image and recomputed the affine region in the rectified image. However, this is expensive as it needs to be done per feature. An alternative is to virtually warp the AC so that the reference view is orthorectified. However, this warping cannot be done exactly, and, as we show in our experiments, this approximate warping leads to decreasing accuracy in the DPR solution as the amount of perspective distortion increases.

Collins and Bartoli [13] derived novel constraints and a minimal solution for the absolute pose of a camera given four co-planar PCs. They first calculate the homography between the object plane and the camera image. While the homography could be directly decomposed into the camera pose, they propose an alternative solution which addresses the spatially-varying accuracy of the homography. They choose an optimal point on the plane to compute a first-order approximation to the homography (an affine transformation) and then compute the pose of the camera based on the 2D point projection and the first-order approximated local transformation. Their solution, called Infinitesimal Plane-based Pose Estimation (IPPE), is therefore highly related to DPR [29], in that it ultimately computes the pose of the camera based on a single AC between the object plane and the image. To apply IPPE in our setting, we generate three virtual PCs around the AC in the reference image by ray intersection with the object plane. However, as we show in our experiments, the IPPE solution is less accurate than ours because of the inherent inaccuracy in the virtual PCs.

Haug and Jühne developed a solution for computing the
absolute pose of a stereo or RGB-D camera from a single AC [22]. Their solution assumes knowledge of the depth of the point in both the reference and query images and thus is not applicable in our scenario of localizing a monocular camera, where the depth of the point in the query image is unknown.

3. Minimal solvers

Notation: We use a sans-serif capital letter \( M \) for a matrix, a bold lower-case letter \( \mathbf{v} \) for a vector, and an italic lower-case letter \( s \) for a scalar. We use subscripts to indicate matrix and vector indexing. A tilde over a vector indicates a homogeneous version of that vector \((\mathbf{x} = [x_1 \ x_2 \ 1]^T)\).

Consider two calibrated pinhole cameras observing a scene: a reference camera, whose pose w.r.t. the world coordinate system is known, and a query camera, whose pose is unknown. Without loss of generality, we assume that the reference camera has extrinsics \([1\ 0\ 0]\), and the query camera has extrinsics \([R\ t]\).\(^1\) Here \(R\) and \(t\) represent the world-to-camera transformation of the query camera, transforming points from the world coordinate system to the coordinate system of the query camera. This is a standard setup in localization pipelines, where our goal is to estimate the 3D rotation \( R \) and translation \( t \) of the query camera.

The inputs to our P1AC method are: corresponding points \( x \) and \( y \) in the reference and query images, respectively; \( A \), the affine transformation from the local region around \( x \) to the local region around \( y \); \( d \), the depth of the observed 3D point in the reference image; and \( n \), the surface normal at the 3D point in the coordinate system of the reference image.

First, we will develop a system of constraints on the absolute pose of the query camera induced by the AC, and then we will describe a minimal solution to this derived system.

3.1. Constraints induced by AC on camera pose

Let us define a function \( f : \mathbb{R}^2 \to \mathbb{R}^2 \) that maps points in the reference image to points in the query image. Let \( x, y \) be corresponding points in the two images such that \( y = f(x) \), and let \( u \) be a point in the neighborhood of \( x \). The first-order Taylor approximation of \( f \) centered at \( x \) is \( f(u) = y + A(u - x) \), where \( A = \nabla_u f(u)|_{u=x} \) is the local affine transformation matrix.

We now aim to analytically derive the form of \( \nabla_u f(u) \) under perspective projection. Let \( \pi^{-1} : \mathbb{R}^2 \to \mathbb{R}^3 \) be the mapping from a point in the reference image to the local 3D surface. Then we have

\[
f(u) = \pi(R\pi^{-1}(u) + t). \tag{1}
\]

The projection function \( \pi(p) \) is straightforward: \( \pi(p) = [p_1/p_3, \ p_2/p_3]^T \). The un-projection function \( \pi^{-1}(u) \), however, requires knowledge of the scene geometry. We assume that the cameras observe a locally planar surface defined by 3D point \( p = dx \) and normal \( n \). To find \( \pi^{-1}(u) \), we intersect ray \( u \) with the plane. The plane is defined by

\[
n^T(p - p') = 0 \tag{2}
\]

where \( p' \) is a point on the planar surface. Parameterizing the ray-plane intersection by \( p' = \alpha \hat{u} \), we have

\[
n^T(\alpha \hat{u} - p) = 0 \tag{3}
\]

and therefore

\[
\alpha = \frac{n^T p}{n^T \hat{u}} \tag{4}
\]

Thus, under the assumption of a locally planar surface, we have determined \( \pi^{-1}(u) \) as follows:

\[
\pi^{-1}(u) = \frac{n^T p}{n^T \hat{u}} \hat{u}. \tag{5}
\]

Let \( q = \pi^{-1}(u) + t \) so that \( f(u) = \pi(q) \). To compute \( \nabla_u f(u) \), we first apply the quotient rule:

\[
\nabla_u f(u) = \nabla_u \pi(q) = \nabla_u \left( \frac{1}{q_3} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \right) = \frac{1}{q_3} \begin{bmatrix} q_3 \nabla_u q_1 - q_1 \nabla_u q_3 \\ q_3 \nabla_u q_2 - q_2 \nabla_u q_3 \end{bmatrix} = \frac{1}{q_3} \nabla_u q_1 - \frac{1}{q_3} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \nabla_u q_3. \tag{6}
\]

To make the expression linear in \( R, t \) and simplify the later solution of the system of equations, we apply the substitution \( v = \frac{1}{q_3} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \) to obtain

\[
\nabla_u f(u) = \frac{1}{q_3} \nabla_u q_1 - v(\nabla_u q_3). \tag{7}
\]

Noting that \( q = \alpha(R\hat{u}) + t \), to compute \( \nabla_u q \) we apply the product rule:

\[
\nabla_u q = (R\hat{u}) \nabla_u \alpha + \alpha \nabla_u (R\hat{u}). \tag{8}
\]

To compute \( \nabla_u \alpha \) we apply the quotient rule:

\[
\nabla_u \alpha = \nabla_u \left( \frac{n^T p}{n^T \hat{u}} \right) = -\frac{(n^T p)(n^T u_{1:2})}{(n^T u)^2} = -\alpha \frac{n^T u_{1:2}}{n^T \hat{u}}. \tag{9}
\]

Note also that \( \nabla_u (R\hat{u}) = R_{1:2} \). Now we plug our expressions for \( \nabla_u \alpha \) (Eq. 9) and \( \nabla_u (R\hat{u}) \) into Eq. 8 to obtain

\[
\nabla_u q = (R\hat{u})(-\alpha \frac{n^T u_{1:2}}{n^T \hat{u}}) + \alpha (R_{1:2} - (R\hat{u}) \frac{n^T u_{1:2}}{n^T \hat{u}}) \tag{10}
\]

\[
= \alpha \left( R_{1:2} - (R\hat{u}) \frac{n^T u_{1:2}}{n^T \hat{u}} \right) \tag{11}
\]

\[
= \frac{n^T p}{n^T \hat{u}} \left( R_{1:2} - (R\hat{u}) \frac{n^T u_{1:2}}{n^T \hat{u}} \right). \tag{12}
\]
Evaluating this expression at \( u = x \) we obtain
\[
\nabla_u q|_{u=x} = d \left( R_{1:1,2} - (Rx) \frac{n_{1:2}^T}{n_1^T} \right). \tag{13}
\]

Finally, we plug Eq. 13 into Eq. 7 to arrive at a formula for \( J = \nabla_u f(u)|_{u=x} \):
\[
J = \frac{d}{m} \left( R_{1:2,1:2} (n_1^T x) - (R_{1:2,2} x) n_{1:2} - \nu (R_{3:1,2} (n_1^T x) - (R_{3,2} x) n_{1:2}) \right) \tag{14}
\]
where \( m = n_1^T x (d(R_{3,2} x) + t_3) \).

We multiply \( m \) on both sides of the constraint \( A \cdot \mathbf{J} = \mathbf{J} \) to cancel out the denominator and make the constraint linear. Our set of six constraints to be solved is thus:
\[
\begin{align*}
y_1(R_3, p + t_3) - (R_1, p + t_1) &= 0, \tag{15} \\
y_2(R_3, p + t_3) - (R_2, p + t_2) &= 0, \tag{16} \\
m_{a_{11}} - m_{j_{11}} &= 0, \tag{17} \\
m_{a_{12}} - m_{j_{12}} &= 0, \tag{18} \\
m_{a_{21}} - m_{j_{21}} &= 0, \tag{19} \\
m_{a_{22}} - m_{j_{22}} &= 0. \tag{20}
\end{align*}
\]
where the first two constraints enforce that the 3D point projects to the 2D observation in the query camera, and the remaining constraints enforce that the affine transformation matrix equals the Jacobian matrix derived above. Note that the only unknowns in equations 14 and 15-18 are elements of \( R \) and \( t \).

### 3.2. Minimal solution

We have six independent constraints on the entries of \( R \) and \( t \) and six degrees of freedom (three for the rotation and three for the translation). Therefore, we can solve for the absolute pose with a single AC.

We use the Cayley parameterization of the rotation with parameters \( x, y, z \) as follows:
\[
R = \frac{1}{s} \begin{bmatrix}
1 + x^2 - y^2 - z^2 & 2(xy + z) & 2(yz - x) \\
2(xy - z) & 1 - x^2 + y^2 - z^2 & 2(xz + y) \\
2(xz - y) & 2(xy + z) & 1 - x^2 - y^2 + z^2
\end{bmatrix} \tag{19}
\]
where \( s = 1 + x^2 + y^2 + z^2 \). Note that we cannot represent 180° rotations with this parameterization; however, this degenerate configuration is not a problem in practice since it would mean that the reference and the query camera have no visual overlap. After plugging this parameterization of \( R \) into equations 15-18, we multiply all equations by \( s \) to cancel out the denominator in \( R \).

Let \( \mathbf{c} = [st, s2, st, x^2, xy, xz, y^2, yz, z^2, x, y, z, 1]^T \). Writing the equations 15-18 as a matrix-vector multiplication \( \mathbf{c} \mathbf{e} = 0 \), we first eliminate \( st \) using Gauss-Jordan elimination. This leaves three quadratic equations in ten monomials on \( x, y, z \) — a 3Q3 problem. Kukelova et al. [30] introduced a fast solver for the 3Q3 problem which produces up to eight solutions for \( x, y, z \). Once we obtain solutions for \( x, y, z \), we solve for \( t \) by backsubstitution into equations 15-18.

### 3.3. Alternative solvers

We also consider two alternative solvers based on previous work, to establish a baseline comparison for the PIAC method proposed in this paper.

#### 3.3.1 Differential pose resection

The Differential Pose Resection (DPR) method [29] solves the PIAC problem under the restriction that the reference image is an ortho-rectified view of the plane. To apply the DPR method in the general case, we need to virtually warp the AC so that it comes from an ortho-rectified view.

Let \( \mathbf{w} \) represent the 2D coordinates of a point on the plane and let \( \mathbf{H} \) be a homography from the plane to the reference image s.t. \( \mathbf{u} = \pi(H\mathbf{w}) \) and \( \mathbf{x} = \pi(H[0 0 1]^T) \). The point corresponding to \( \mathbf{u} \) is \( \mathbf{v} = g(\mathbf{w}) = f(\pi(H\mathbf{w})) \). \( g \) is thus a non-linear transformation from the plane to the query. We replace \( f \) with \( \hat{f} \) since we only know the AC between the reference and query images. We can approximate \( g \) as an affine transformation by computing the matrix \( \mathbf{A}' = \nabla_w g/\hat{g} \):
\[
\mathbf{A}' = \begin{bmatrix}
H_{11} & (H_{13} H_{11})/H_{33} & H_{12} - (H_{13} H_{23})/H_{33} \\
H_{21} & (H_{23} H_{21})/H_{33} & H_{22} - (H_{23} H_{32})/H_{33}
\end{bmatrix}.
\]

#### 3.3.2 Infinitesimal plane-based pose estimation

The Infinitesimal Plane-based Pose Estimation (IPPE) method [13] solves for the pose of the query camera given four coplanar PCs. In our case, the AC only contains a single PC. To transform a PIAC problem into an IPPE problem, we generate three extra virtual PCs by sampling 2D points around \( \mathbf{x} \) in the reference image: the 3D points are found by ray intersection with the plane (Eq. 5), and the 2D points are found by applying the affine matrix \( \mathbf{A} \). We generate three extra virtual PCs in this manner and provide them along with the PC from the original AC as input to IPPE. It is important to note that these virtual correspondences are only approximations generated by the assumption that the AC lies on an actual plane, and that the AC is valid beyond an infinitesimal region around the 2D point.

### 4. Evaluation

We performed evaluations on synthetic and real data to test the numerical stability of our solver, test robustness to various types of noise, and assess its performance as part of an image-based localization system.

We compared the following algorithms in our evaluation:

- **P3P** Standard P3P absolute pose calculation from three 2D-3D point observations. We used the Lambda Twist P3P solver [44].
- **PIAC**: Our solution to the PIAC problem.
• DPR: Differential Pose Resection [29].
• IPPE: Infinitesimal Plane Pose Estimation [13].

In the experiments, we computed two pose accuracy metrics. Given the true camera rotation $\mathbf{R}$ and the estimated camera rotation $\hat{\mathbf{R}}$, we computed the rotation error as $|| \ln(\mathbf{R}\hat{\mathbf{R}}^T)||$, where $\ln(\cdot)$ denotes the SO(3) logarithm [23, 21]. While $\arccos((\text{trace}(\mathbf{R}) - 1)/2)$ is often used as a rotation error metric, we instead use a robust implementation of $|| \ln(\mathbf{R}\hat{\mathbf{R}}^T)||$, which we found to be more stable for small rotations. Given the true camera center $\mathbf{c}$ and the estimated camera center $\hat{\mathbf{c}}$, we computed the position error as $||\mathbf{c} - \hat{\mathbf{c}}||$. In the synthetic experiments, when considering multiple solutions produced by a method, we choose the solution that minimizes the max. of these two error metrics.

4.1. Synthetic data experiments

We generated synthetic problem instances with varying levels of noise added to the 2D point observations, affine transformations, and normal vectors.

We generated synthetic random problems in the following manner adapted from [15]. We randomly place two cameras around the origin at a distance sampled from $[-0.5, 0.5]^3$. To generate a correspondence between the two cameras, we select a random 3D point from $\mathcal{N}(0, I_{3\times3})$ with a random normal vector. We project the 3D point to the two cameras using a focal length of 400. Then we calculate the affine transformation from the local homography between the first and second cameras [4]. We transform the cameras and 3D points so that the first camera (the reference camera) is at the origin with identity rotation.

We add Gaussian noise to the 2D point observations. Following prior studies [16, 15], we add Gaussian noise to the elements of the $2 \times 2$ affine transformation. We select the level of affine noise based on the chosen percentage error; i.e., if the chosen percentage error is $e$, the std. dev. of the noise added to $a_{ij}$ is $e \cdot a_{ij}$. Finally, we add noise to
The value at lying elliptic equations in the P3P problem which are solved the rotation and translation parameters to reveal the under-
is about 2.73 $\mu s$. The results are given in Table 1.

### Table 1: Average timing in $\mu s$ over 10,000 trials.

<table>
<thead>
<tr>
<th>Method</th>
<th>P3P</th>
<th>P1AC</th>
<th>DPR</th>
<th>IPPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time ($\mu s$)</td>
<td>0.54</td>
<td>2.73</td>
<td>3.23</td>
<td>25.27</td>
</tr>
</tbody>
</table>

The synthetic experiments show that the P1AC solver is substantially outperforms P3P (see Section 4.2).

### Noise

We tested the P1AC solver and the P3P solver over a range of noise levels in the 2D point observations, affine transformations, and surface normals. We tested 1,000 random problem instances for each noise level. Figure 2b plots the median errors for each method at each setting. The chosen noise ranges are the same as used in [15]. The left column presents the results for increasing levels of point and affine noise, with the normal noise set to 1 degree. The center column presents the results for increasing levels of point and normal vector noise, with the affine noise set to 4% [15]. The right column presents the results for the P3P solver, for comparison. The P3P solver does not use the affine transformation or normal vector and thus is unaffected by affine and normal noise.

Timings. We computed the average processing times of the solvers over 10,000 random problem instances. The solvers were all implemented in C++ and timings were made on an Apple M1 Pro CPU. We used the implementations of Lambda Twist and 3Q3 provided in PoseLib [31] and the implementation of IPPE provided in OpenCV’s solvePnP function. The results are given in Table 1.

Among the 1AC variants, P1AC is the fastest with an average timing of 2.73 $\mu s$. The Lambda Twist P3P solver is about 6 $\times$ faster than P1AC. Note that this implementation of the P3P solver is based on a careful elimination of the rotation and translation parameters to reveal the underlying elliptic equations in the P3P problem which are solved through diagonalization.

Robust estimation. We investigated the performance of P1AC and P3P when used in robust estimation with an increasing outlier ratio, and confirmed that within a robust estimation loop, P1AC outperforms P3P in terms of speed.

We tested a range of outlier ratios from 0 to 0.9, with 10,000 trials at each setting. In each trial we randomly generate 1,000 correspondences with a point noise set at 1 pixel, affine noise at 4%, and normal noise at 1$. We replace a portion of the correspondences with random values according to the desired outlier ratio. We then run vanilla RANSAC [17] or LO-RANSAC [32] using either P1AC or P3P as the minimal solver and EPnP [33] as the non-minimal solver in local optimization. We tested vanilla RANSAC to analyze the trade-off in our approach between having a smaller minimal sample size but greater noise sensitivity. LO-RANSAC mitigates noise sensitivity in the minimal solver by incorporating local optimization (LO) iterations to grow the inlier set from the initial estimate.

The average error was stable for each method across the range of outlier ratios. Table 2 shows the average error for each method across all outlier ratios. When employing vanilla RANSAC, P1AC is less accurate than P3P at this level of noise. When employing LO-RANSAC, the error for P1AC is marginally higher than that for P3P, the error rate for both methods remains extremely low, attesting to their effectiveness. Note that on real data, P1AC substantially outperforms P3P (see Section 4.2).

Figure 4 plots the average timing for each method. When
using vanilla RANSAC, P1AC is faster than P3P past an outlier ratio of about 0.45; with LO-RANSAC, P1AC is faster across all outlier ratios. This result indicates that our P1AC solver is accurate enough to provide a good initialization for LO and reach convergence more quickly than P3P, even with a low outlier ratio. In our real data experiments (Section 4.2), to be consistent with the state-of-the-art, we use Graph Cut RANSAC (GC-RANSAC) [5], which improves upon LO-RANSAC by incorporating graph cut optimization into the LO iterations. We did not use GC-RANSAC in our synthetic data experiments because the observations, being randomly generated, did not exhibit spatial coherence.

Mirrored solutions and degeneracies. The solutions returned by our P1AC solver include mirrored solutions, meaning pairs of solutions where the cameras are rotated exactly 180 degrees from each other. Previous work on plane-based pose estimation has also mentioned the existence of mirrored solutions [13]; they are indeed common for minimal problems and are an artifact of not including inequalities in the constraints. In practice they are not a problem since such solutions are filtered out inside RANSAC.

In addition to the 180° degeneracy inherent to the Cayley rotation representation, we noticed a degeneracy for rotations very close to identity when using the PoseLib 3Q3 solver implementation [31]. However, when using a 3Q3 solver produced by the GAPS automatic generator [34], this degeneracy for near-identity rotations does not appear. Another observation we made is that the translation accuracy of both P1AC and P3P degrades in weak-perspective situations, where the plane is very far from the camera. These aspects are discussed further in the supplemental material.

4.2. Real data experiments

To evaluate the performance of our method for large-scale image-based localization, we used the Cambridge Landmarks [26] and Aachen Day-Night v1.1 [47, 46, 55] benchmark datasets. Both datasets are commonly used in the image-based localization literature.

The Cambridge Landmarks dataset consists of six scenes, each recorded through multiple video sequences taken by a mobile phone, depicting parts of Cambridge, UK. Out of the sequences recorded for each dataset, some are used to obtain database images that represent the scene while the others are used to obtain query images. Ground truth poses and intrinsic camera calibrations for all images were obtained using the VisualSFM [52, 53] Structure-from-Motion (SfM) software. Image-based localization performance is typically measured by reporting the median position and orientation error. In addition, we measure the percentage of images localized within 10cm/1° and 20cm/1° of their ground truth poses.

Cambridge Landmarks is a relatively easy dataset since each scene is rather small and the query images were taken around the same time as the database images. A much more challenging case is the Aachen Day-Night dataset, which depicts the historical inner city of Aachen, Germany. The database images were all taken during daytime conditions and the dataset provides query images taken at nighttime conditions over a longer period of time. Ground truth for the database images were obtained via COLMAP [48]. The nighttime queries were later registered by refining initial pose estimates [55]. We follow the common evaluation protocol and report the percentage of images localized within three error thresholds (0.25m/2°, 0.5m/5°, 5.0m/10°).

There are multiple ways to obtain affine features from real images. The most standard is to use a local feature detector, like DoG [35] or Key.Net [7], estimate keypoint locations and scales, and use the patch-based AffNet [39] to get affine shapes. Finally, a patch-based descriptor, like HardNet [38] or SOSNet [49], is applied. In our experiments, we combine DoG for feature detection, AffNet for affine shape recovery, and HardNet for patch description, an approach that is among the leaders in the IMC 2020 benchmark [24]. Given a SfM point cloud, we apply the normal estimator implemented in MeshLab to obtain an oriented point cloud. The normals are fitted to the 200 nearest neighbors.

Image localization procedure. For both datasets, we use image retrieval to identify a small set of potentially relevant database images for each query photo. We use DenseVLAD [50] descriptors for Cambridge Landmarks and NetVLAD [2] descriptors for Aachen Day-Night. We then establish 2D-2D correspondences between each query

<table>
<thead>
<tr>
<th>Robust estimator</th>
<th>P1AC</th>
<th>P3P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rot. (°)</td>
<td>Pos.</td>
<td>Rot. (°)</td>
</tr>
<tr>
<td>RANSAC</td>
<td>0.450</td>
<td>0.019</td>
</tr>
<tr>
<td>LO-RANSAC</td>
<td>0.085</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table 2: Average rotation and pose error for each robust estimation method on synthetic data over a range of outlier ratios.
Table 3: Cambridge Landmarks [26] median position (centimeters) and rotation (degrees) errors, and recalls (percentages), at 0.1m/1° and 0.2m/1°, of GC-RANSAC [5] combined with various solvers. The average over all scenes and average weighted by number of images in each scene are in the last row.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Position (cm) ↓</th>
<th>Rotation (°) ↓</th>
<th>Recall (0.1m/1°) ↑</th>
<th>Recall (0.2m/1°) ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Great Court</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>King’s College</td>
<td>10</td>
<td>8</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Old Hospital</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Shop Facade</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>St Mary’s Church</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Street</td>
<td>32</td>
<td>50</td>
<td>248</td>
<td>20</td>
</tr>
<tr>
<td>Avg.</td>
<td>12</td>
<td>13</td>
<td>47</td>
<td>8</td>
</tr>
<tr>
<td>Weighted avg.</td>
<td>22</td>
<td>33</td>
<td>153</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 4: Aachen Day-Night pose error recalls [47], in percentages, at 0.25m/2°, 0.5m/5°, and 5.0m/10° of GC-RANSAC [5] combined with various solvers.

<table>
<thead>
<tr>
<th>Scene</th>
<th>Recall (0.25m/2°) ↑</th>
<th>Recall (0.5m/5°) ↑</th>
<th>Recall (5m/10°) ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day</td>
<td>62.0</td>
<td>62.0</td>
<td>62.0</td>
</tr>
<tr>
<td>Night</td>
<td>47.1</td>
<td>46.1</td>
<td>51.3</td>
</tr>
<tr>
<td>Day</td>
<td>83.4</td>
<td>83.4</td>
<td>84.6</td>
</tr>
<tr>
<td>Night</td>
<td>60.2</td>
<td>63.9</td>
<td>66.0</td>
</tr>
<tr>
<td>Day</td>
<td>96.0</td>
<td>95.6</td>
<td>96.0</td>
</tr>
<tr>
<td>Night</td>
<td>74.3</td>
<td>75.9</td>
<td>76.4</td>
</tr>
</tbody>
</table>

and its retrieved database images. These 2D-2D matches are then lifted to 2D-3D correspondences. For Cambridge Landmarks, we use the 3D points in SfM point clouds corresponding to the features extracted from the database images. For Aachen, we use a dense depth maps obtained by rendering a mesh model of the scene to obtain the 3D points (and their corresponding normals) for the database features [43].

In these experiments we aim to analyze the usefulness of the P1AC solver in practice. To this end we use advanced RANSAC techniques and tune them per solver, to achieve the best performance possible for each method.

To estimate the pose of each query image, we iterate through the retrieved database images. Between each pair, we estimate the pose by GC-RANSAC [5] using the implementation provided by the authors. The final pose for a query image comes from the database image with which the highest inlier number is achieved. GC-RANSAC requires two types of solvers, one for minimal and one for non-minimal pose estimation. We test the proposed P1AC solver as the minimal solver and compare it to the P3P solver [44]. The non-minimal solver, running both in the local optimization and in the final refitting on all inliers, is always EPnP [33] followed by a few iterations of the Levenberg-Marquardt optimization [40] minimizing the reprojection error. The proposed P1AC solver enables running GC-RANSAC exhaustively, without severely affecting the run-time. Therefore, we perform an exhaustive search over all correspondences without using a random sampler. For P3P solver, we use the PROSAC sampler [11] on matches ordered by SNN ratio.

For Cambridge Landmarks, we set the inlier-outlier threshold to 4 pixels. For Aachen Day-Night, we set it to 12 pixels. We tuned the SNN ratio threshold for both P3P and P1AC. For P3P, 0.9 works the best. Surprisingly, P1AC and other 1AC solvers are the most accurate on Cambridge Landmarks without any SNN ratio filtering. Even though this leads to slightly increased run-times due to running on much more points, it increases both the accuracy and inlier number significantly.

**Pose accuracy.** A summary of errors on the Cambridge Landmarks dataset is given in Table 3. We report the median position (in meters) and rotation errors (in degrees), and the recall (in percentages) at 0.1m/1° and 0.2m/1°. The proposed P1AC method has lower errors and thus, higher recall on all scenes. In many cases, the recall increase is significant compared to the P3P solver. For example, on scene Street (the most challenging scene in the dataset) both recall values increase by around 10 percentage points (pp). The average 0.1m/1° over all scenes is increased by 10.5pp and the average 0.2m/1° by 7.6pp. Compared to DPR and IPPE, the proposed P1AC solver increases the recall by 1-3pp on all scenes, while significantly reducing the median position error.

The recalls at 0.25m/2°, 0.5m/5°, and 5.0m/10° on the Aachen Day-Night [47] dataset are in Table 4. In this case, all methods use an SNN ratio threshold set to 0.9 since increasing it led to a deterioration in the accuracy. This is likely caused by the dataset being more challenging than the Cambridge Landmarks. While the accuracy difference is smaller than in Table 3, the proposed P1AC consistently
Improves all metrics on the Night sequence and leads to similar accuracy to all other methods in the Day sequence.

**Timing.** Table 5 reports the average processing times (in seconds) of all compared methods. On Cambridge Landmarks, as expected from the excluded SNN ratio filtering and the exhaustive search used for 1AC methods, their runtime is larger than that of P3P. On Aachen Day-Night, all methods use the same SNN ratio threshold and the proposed P1AC is faster on both scenes than P3P. P1AC is the fastest on average on the Day sequence, and runs at a comparable speed to other methods in the Night sequence.

In our experiments, we used the same set of matches (DoG+HardNet+AffNet) for all methods, with the P3P method disregarding the affine components. As such, no method had a timing advantage related to feature extraction. However, for completeness, we measured feature extraction time and found that DoG+HardNet needs 0.18 ms per feature, and DoG+HardNet+AffNet 0.53 ms. Note that AffNet plays a crucial role in maintaining high accuracy and is not solely used for generating affine features [24].

### 4.3. Discussion

In terms of the timing of the minimal solver by itself, our P1AC solver is slower than the P3P solver. However, our P1AC solver is still quite fast (< 3μs) and moreover the minimal solver computation time is usually negligible when compared to other steps in the random sampling process such as local optimization and inlier testing.

While our P1AC solver itself is slightly slower than P3P, and is susceptible to more types of noise, it has a number of beneficial properties due to estimating the pose from a single AC, such as deterministic and fast robust estimation. On real data, we show that, when employed inside a robust estimator, our P1AC solver produces more accurate results and also is faster in some cases (e.g., Aachen Day-Night).

In comparison to alternative baseline methods, DPR and IPPE, our P1AC method successfully localizes more frames and is more accurate. This indicates that our proposed approach, based on differential analysis of the AC, more accurately models the constraint imposed by an AC on the absolute pose of the query camera.

Improving the AC quality [6] might lead to a speed up in sample consensus because the initial pose estimates by the P1AC solver would be more accurate, so the system would spend less time on LO steps. Other minimal solvers that use a single correspondence often use histogram or kernel density voting in place of RANSAC, although they typically estimate a single parameter such as the focal length [10, 19]. Future work lies in analyzing how ACs and the P1AC constraints could be applied to 6DOF camera pose voting [54, 1]. While we only use point reprojection error for inlier testing due to the inherent noise in the affine components [6], future work could include testing first-order compatibility.

### 5. Conclusions

In this work, we derived novel constraints imposed by an affine correspondence on the absolute pose of a camera, given knowledge of a 3D point in the scene, the surface normal at that point, and the pose of the reference image. Using these novel constraints we developed a minimal solution for absolute pose from a single AC. Through experiments on synthetic data we established the numerical stability of our solver and its behavior under increasing levels of noise. We demonstrated the usefulness of our solver in an image-based localization system. In our experiments, replacing P3P with our P1AC solver improved localization accuracy and recall.

**Acknowledgements.** This work was supported by NSF Award No. 2144822, EU Horizon 2020 project RICAIP (No. 857306), the European Regional Development Fund under project IMPACT (No. CZ.02.1.01/0.0/0.0/15 003/000468), and the Czech Science Foundation (GAČR) JUNIOR STAR Grant No. 22-23183M. D. Barath was supported by the ETH Postdoc Fellowship.

### References


