Affine Perspective-Three-Point Problem

Gaku Nakano
NEC Corporation, 1753 Shimonumabe, Kawasaki, Japan
g-nakano@nec.com

Abstract

This paper addresses the Perspective-Three-Point (P3P) problem under affine camera models. We derive direct closed-form solvers for weak perspective and para perspective, which are representative affine camera models. The affine P3P solution reduces to a bi-quadratic equation. Unlike exact P3P solvers that require a cubic or quartic equation, it allows for the simple and stable calculation of real solutions using the quadratic formula. Since affine approximations are valid only when scene depth variation is small, we further propose an iterative correction that upgrades the affine solution to the exact P3P solution. Through extensive comparisons using synthetic data and public datasets, we demonstrate that affine P3P solvers with two upgrade iterations achieve performance substantially comparable to that of the state-of-the-art P3P solver.

1. Introduction

The Perspective-three-Point (P3P) problem—the minimal formulation for calibrated camera pose estimation from three 2D-3D point correspondences—has been a classical and fundamental topic for a wide range of vision tasks, including Structure-from-Motion (SfM), Visual SLAM, and robot navigation. The classical derivation treats the distances from the camera to the 3D points as unknowns, organizes the law of cosines between the points, and reduces the problem to a quartic polynomial [16, 18]. In the early days, singular value decomposition was further used to calculate the camera pose after solving the quartic equation. Its properties, such as at most four real solutions and degeneracy conditions, are thoroughly investigated [15]. Later, direct solvers without singular value decomposition have been widely studied [24, 26, 33]. Furthermore, the P3P problem has been reinterpreted as a problem of the intersection of two conics [12, 36], which leads to formulations that combine a cubic and a quadratic equations and improves numerical stability and computational efficiency. Primary research interests for P3P solvers, as a minimal solver within RANSAC [10, 13], are numerical stability and speed.

Many derivatives and extensions build on the P3P problem. P4Pf considers unknown focal length of uncalibrated cameras [6, 44], and P4Pfr/P5Pfr additionally handle unknown radial distortions [7, 27, 32]. By exploiting affine- or scale-invariant features, P1P [42] and P2P [43] reduce the number of required correspondences to one or two points. The PnP problem is the task for finding camera pose in the least squares case ($N > 3$) [40, 45, 46]. In recent years, learning-based PnP solvers have also been explored [3, 8].

Perspective projection is inherently non-linear, yet under certain imaging conditions and scene geometries, affine camera models can be more appropriate alternative. For example, far-field imaging where scene depth variation is small relative to the distance to the camera. Among affine models, weak perspective and para perspective are representative approximations. Alter [2] showed that the motion of weak perspective cameras can be derived from three point correspondences as in the P3P problem. Ohta *et al*. [35] proposed para perspective projection to enhance affine camera approximation. Historically, affine assumptions underpinned classical frameworks in SfM, such as factorization methods [37, 41]; however, as large-scale SfM pipelines have been established [1, 38], research interests in affine models decreased. For $N > 3$ case, several solvers based on weak perspective were proposed [11, 17, 29, 34, 39], but as fully perspective PnP solvers matured, affine-based methods have been less attention.

This paper revisits the P3P problem from the above broader historical and practical perspectives. We reveal a unified treatment of affine P3P: both weak and para perspective share the same derivation—centroid shift and nullspace parameterization—and collapse to a single bi-quadratic equation. The only model-specific step is a simple back-substitution to recover the camera motion. We further propose a lightweight upgrade procedure that, starting from an affine solution, reaches the exact P3P solution in a few iterations. This provides a smooth bridge between affine and perspective solutions. Our focus is not to enter affine cameras into P3P benchmark competitions, but to demonstrate their value in simplifying geometric problems.

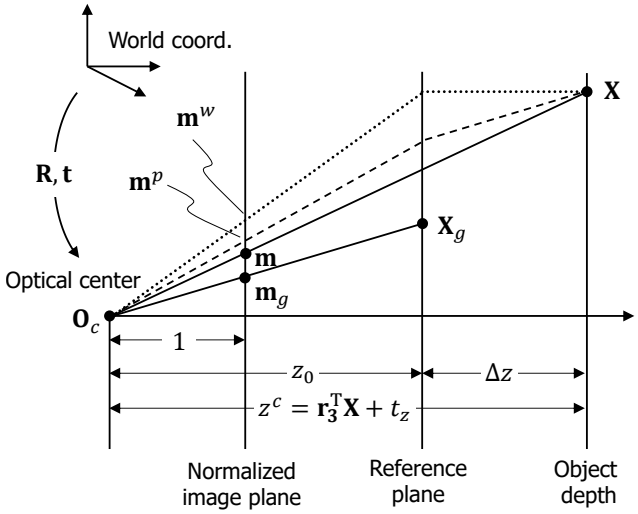

Figure 1. Perspective and affine projection models. A 3D point $\mathbf{X}$ is observed as $\mathbf{m}$ in the normalized image plane by perspective projection ($\mathbf{X}$—$\mathbf{m}$). Weak perspective orthogonally projects $\mathbf{X}$ in the reference plane on the centroid $\mathbf{X}_g$, then scales it to $\mathbf{m}^w$ in the image plane ($\mathbf{X} \cdots \mathbf{m}^w$). Para perspective projects $\mathbf{X}$ in the direction parallel to $\overrightarrow{\mathbf{O}_c\mathbf{X}_g}$ on the reference plane, then scales it to $\mathbf{m}^p$ in the image ($\mathbf{X}$- - -$\mathbf{m}^p$).

2. Perspective and affine projections

This section reviews the theory of perspective projection under the pinhole camera model and its affine approximations, namely weak perspective and para perspective. Throughout the paper, cameras are assumed to be calibrated; intrinsic parameters such as focal length and lens distortions are known. Thus, all image measurements are expressed in the normalized image coordinates. Readers may refer to [2, 5, 21, 22, 35] for details of affine projections.

2.1. Perspective

As illustrated in Fig. 1, let a 3D point $\mathbf{X}_i = [x_i, y_i, z_i]^\mathsf{T}$ be observed as the image point $\mathbf{m}_i = [u_i, v_i]^\mathsf{T}$ by a pinhole camera with rotation $\mathtt{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^\mathsf{T}$ and translation $\mathbf{t} = [t_x, t_y, t_z]^\mathsf{T}$. The projective equation is given by

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \frac{1}{z_i^c} \left(\begin{bmatrix} \mathbf{r}_1^\mathsf{T} \\ \mathbf{r}_2^\mathsf{T} \\ \mathbf{r}_3^\mathsf{T} \end{bmatrix} \mathbf{X}_i + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \right) \leftrightarrow \begin{bmatrix} \mathbf{m}_i \\ 1 \end{bmatrix} = \frac{1}{z_i^c} (\mathtt{R}\mathbf{X}_i + \mathbf{t}) \tag{1}$$

where z_i^c denotes the depth of $\mathbf{X}_i$ measured from the optical center, *i.e.*,

$$z_i^c = \mathbf{r}_3^\mathsf{T}\mathbf{X}_i + t_z. \tag{2}$$

Equation (1) can be rewritten in an inhomogeneous form:

$$\mathbf{m}_i = \frac{1}{z_i^c} \begin{bmatrix} \mathbf{r}_1^\mathsf{T}\mathbf{X}_i + t_x \\ \mathbf{r}_2^\mathsf{T}\mathbf{X}_i + t_y \end{bmatrix}. \tag{3}$$

Since z_i^c varies across each 3D point, perspective projection is a non-linear mapping due to the division by z_i^c.

2.2. Weak perspective

Affine camera models remove the non-linearity of perspective projection. Among them, weak perspective is the zeroth-order approximation [2, 21, 22] and also known as scaled-orthographic.

Weak perspective first orthographically projects $\mathbf{X}_i$ onto a reference plane that is parallel to the image plane and passes through the object centroid $\mathbf{X}_g$, and then projects it onto the image plane (See the dotted line $\mathbf{X} \cdots \mathbf{m}^w$ in Fig. 1). Let z_0 be the distance from the optical center to the reference plane. The two-step mapping can be written by

$$\mathbf{m}_i^w = \frac{1}{z_0} \begin{bmatrix} \mathbf{r}_1^\mathsf{T}\mathbf{X}_i + t_x \\ \mathbf{r}_2^\mathsf{T}\mathbf{X}_i + t_y \end{bmatrix}. \tag{4}$$

Weak perspective approximates the depth of every 3D point by a constant $z_i^c \approx z_0$, regardless of the true depth z_i^c.

2.3. Para perspective

Para perspective, first developed by Ohta *et al.* [35], is a more accurate affine camera model that is a first-order approximation to perspective projection [5, 21]. Unlike weak perspective, it first projects 3D points onto the reference plane along the direction parallel to the line through the optical center and the object centroid $\mathbf{X}_g$ (See the dashed line $\mathbf{X}$- - -$\mathbf{m}^p$ in Fig. 1).

Let us define $x_0 = (\mathbf{r}_1^\mathsf{T}\mathbf{X}_g + t_x)/z_0$ and $y_0 = (\mathbf{r}_2^\mathsf{T}\mathbf{X}_g + t_y)/z_0$. The para perspective can be mapped by

$$\mathbf{m}_i^p = \frac{1}{z_0} \begin{bmatrix} (\mathbf{r}_1 - x_0\mathbf{r}_3)^\mathsf{T}\mathbf{X}_i + t_x \\ (\mathbf{r}_2 - y_0\mathbf{r}_3)^\mathsf{T}\mathbf{X}_i + t_y \end{bmatrix} \tag{5}$$

As in weak perspective, the object depth is approximated by a constant $z_i^c \approx z_0$, but the correction terms involving $\mathbf{r}_3$ of the rotation matrix $\mathtt{R}$ improve the approximation accuracy.

2.4. Properties of affine projections

Due to the constant depth assumption, neither weak nor para perspective projection produces vanishing points: parallel lines in 3D space remain parallel in 2D images. This assumption holds when the depth deviation $\Delta z = |z_i^c - z_0|$ is sufficiently small relative to z_0. Since the affine projections keep linearity, the projection of the object centroid $\mathbf{X}_g$ coincides with the centroid of the image points $\mathbf{m}_g$, *i.e.*, $(x_0, y_0) = (u_g, v_g)$.

The affine approximation is most accurate when $\mathbf{X}_g$ lies on the optical axis, and the projection errors, $\|\mathbf{m} - \mathbf{m}^w\|$ and $\|\mathbf{m} - \mathbf{m}^p\|$, increase as $\mathbf{X}$ and $\mathbf{X}_g$ move away from the optical axis. In other words, even for 3D points at the same depth, the projection error grows with distance from the optical center, or close to the image boundary.

3. Affine perspective-three-point problem

This section presents direct minimal solvers for the affine P3P problem under weak perspective and para perspective. In addition, it describes an iterative refinement that employs the affine P3P solution as an initial guess and enforces the perspective projection constraint. The complete algorithm is summarized in the supplementary material.

3.1. Weak perspective

As described in Sec. 2.4, since the projection of the object centroid $\mathbf{X}_g = \frac{1}{3}\sum \mathbf{X}_i$ coincides with the image point centroid $\mathbf{m}_g = \frac{1}{3}\sum \mathbf{m}_i$, the translation $\mathbf{t}$ can be canceled by subtracting their centroid from the points: $\hat{\mathbf{m}}_i^w = \mathbf{m}_i^w - \mathbf{m}_g$ and $\hat{\mathbf{X}}_i = \mathbf{X}_i - \mathbf{X}_g$. The centroid shifting also moves the world origin to the object centroid, where the new translation becomes $z_0[u_g, v_g, 1]^\mathsf{T}$ that is equivalent to $\mathbf{t} + \mathtt{R}\mathbf{X}_g$ in the original world coordinates.

Define $\mathbf{p} = \frac{1}{z_0}\mathbf{r}_1$ and $\mathbf{q} = \frac{1}{z_0}\mathbf{r}_2$. From Eq. (4), we can obtain

$$\begin{bmatrix}\hat{\mathbf{m}}_1^w & \hat{\mathbf{m}}_2^w\end{bmatrix} = \begin{bmatrix}\mathbf{p}^\mathsf{T} \\ \mathbf{q}^\mathsf{T}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{X}}_1 & \hat{\mathbf{X}}_2\end{bmatrix}. \tag{6}$$

Note that $\hat{\mathbf{m}}_3$ and $\mathbf{X}_3$ are excluded due to the linearity. Since the left-hand side matrix is 2×2, we obtain

$$\begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} = \begin{bmatrix}\mathbf{p}^\mathsf{T} \\ \mathbf{q}^\mathsf{T}\end{bmatrix}\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}, \tag{7}$$

where

$$\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2\end{bmatrix} = \begin{bmatrix}\hat{\mathbf{X}}_1 & \hat{\mathbf{X}}_2\end{bmatrix}\begin{bmatrix}\hat{\mathbf{m}}_1^w & \hat{\mathbf{m}}_2^w\end{bmatrix}^{-1}. \tag{8}$$

Equation (7) shows the following four constraints with respect to $\mathbf{p}$ and $\mathbf{q}$:

$$\begin{aligned}\mathbf{p}^\mathsf{T}\mathbf{a}_1 = 1, \quad \mathbf{p}^\mathsf{T}\mathbf{a}_2 = 0, \\ \mathbf{q}^\mathsf{T}\mathbf{a}_1 = 0, \quad \mathbf{q}^\mathsf{T}\mathbf{a}_2 = 1.\end{aligned} \tag{9}$$

We can rewrite Eq. (9) in a matrix form as

$$\begin{bmatrix}\mathbf{a}_1^\mathsf{T} & -1 \\ \mathbf{a}_2^\mathsf{T} & 0\end{bmatrix}\begin{bmatrix}\mathbf{p} \\ 1\end{bmatrix} = \begin{bmatrix}0 \\ 0\end{bmatrix}, \quad \begin{bmatrix}\mathbf{a}_1^\mathsf{T} & 0 \\ \mathbf{a}_2^\mathsf{T} & -1\end{bmatrix}\begin{bmatrix}\mathbf{q} \\ 1\end{bmatrix} = \begin{bmatrix}0 \\ 0\end{bmatrix}. \tag{10}$$

The unknown vectors, $[\mathbf{p}^\mathsf{T}, 1]^\mathsf{T}$ and $[\mathbf{q}^\mathsf{T}, 1]^\mathsf{T}$, can be parameterized by a linear combination of the nullspace of the coefficient matrices, *i.e.*,

$$\begin{bmatrix}\mathbf{p} \\ 1\end{bmatrix} = \alpha\mathbf{n}_1 + \gamma\mathbf{n}_2, \quad \begin{bmatrix}\mathbf{q} \\ 1\end{bmatrix} = \beta\mathbf{n}_1 + \delta\mathbf{n}_3, \tag{11}$$

where $\mathbf{n}_i$ represents the nullspace vector, and $\alpha, \beta, \gamma, \delta$ are unknown coefficients. The two 2×4 coefficient matrices in Eq. (10) have the same first 2×3 components, differing only in their fourth column. Therefore, Eq. (11) share the same nullspace vector $\mathbf{n}_1$. Letting a_{ij} be the j-th element of $\mathbf{a}_i$, we can manually derive the nullspace vectors $\mathbf{n}_i$ by

$$\mathbf{n}_1 = \begin{bmatrix}a_{21}a_{32} - a_{31}a_{22} \\ a_{31}a_{12} - a_{11}a_{32} \\ a_{11}a_{22} - a_{21}a_{12} \\ 0\end{bmatrix},$$

$$\mathbf{n}_2 = \begin{bmatrix}a_{22} \\ -a_{12} \\ 0 \\ a_{11}a_{22} - a_{12}a_{21}\end{bmatrix}, \ \mathbf{n}_3 = \begin{bmatrix}-a_{21} \\ a_{11} \\ 0 \\ a_{11}a_{22} - a_{12}a_{21}\end{bmatrix}. \tag{12}$$

To ensure $\{\mathbf{n}_1, \mathbf{n}_2\}$ and $\{\mathbf{n}_1, \mathbf{n}_3\}$ to be orthogonal, we apply the Gram-Schmidt orthogonalization:

$$\begin{aligned}\mathbf{n}_1 \leftarrow \mathbf{n}_1 / \left\|\mathbf{n}_1\right\|, \\ \mathbf{n}_2 \leftarrow \mathbf{n}_2 - (\mathbf{n}_1^\mathsf{T}\mathbf{n}_2)\mathbf{n}_1, \quad \mathbf{n}_3 \leftarrow \mathbf{n}_3 - (\mathbf{n}_1^\mathsf{T}\mathbf{n}_3)\mathbf{n}_1.\end{aligned} \tag{13}$$

The last element of $\mathbf{n}_1$ is $n_{41} = 0$, and Eq. (11) shows that the fourth element of the two unknown vectors is one. Therefore, we normalize $\mathbf{n}_2$ and $\mathbf{n}_3$ by

$$\mathbf{n}_2 \leftarrow \mathbf{n}_2 / n_{24}, \quad \mathbf{n}_3 \leftarrow \mathbf{n}_3 / n_{34}. \tag{14}$$

As a result, we can determine $\gamma = \delta = 1$ and rewrite Eq. (11) by

$$\begin{aligned}\begin{bmatrix}\mathbf{p} \\ 1\end{bmatrix} = \alpha\mathbf{n}_1 + \mathbf{n}_2 = \alpha\begin{bmatrix}\mathbf{v}_1 \\ 0\end{bmatrix} + \begin{bmatrix}\mathbf{v}_2 \\ 1\end{bmatrix}, \\ \begin{bmatrix}\mathbf{q} \\ 1\end{bmatrix} = \beta\mathbf{n}_1 + \mathbf{n}_3 = \beta\begin{bmatrix}\mathbf{v}_1 \\ 0\end{bmatrix} + \begin{bmatrix}\mathbf{v}_3 \\ 1\end{bmatrix}.\end{aligned} \tag{15}$$

Note that the null-vectors are orthogonal but not orthonormal: $\mathbf{v}_1^\mathsf{T}\mathbf{v}_2 = \mathbf{v}_1^\mathsf{T}\mathbf{v}_3 = 0, \|\mathbf{v}_1\| = 1, \|\mathbf{v}_2\| \neq \|\mathbf{v}_3\| \neq 1$.

Now, we consider how to find the two unknown variables α and β. Using the rotation matrix constraints $\mathtt{R}^\mathsf{T}\mathtt{R} = \mathrm{I}$, we can obtain the following two polynomials in α and β:

$$\begin{aligned}\mathbf{p}^\mathsf{T}\mathbf{q} = \begin{cases}\frac{1}{z_0^2}\mathbf{r}_1^\mathsf{T}\mathbf{r}_2 = 0 \\ (\alpha\mathbf{v}_1 + \mathbf{v}_2)^\mathsf{T}(\beta\mathbf{v}_1 + \mathbf{v}_3) = \alpha\beta + \mathbf{v}_2^\mathsf{T}\mathbf{v}_3\end{cases} \\ \rightarrow \alpha\beta + \mathbf{v}_2^\mathsf{T}\mathbf{v}_3 = 0,\end{aligned} \tag{16}$$

$$\begin{aligned}\left\|\mathbf{p}\right\|^2 - \left\|\mathbf{q}\right\|^2 = \\ \begin{cases}\frac{1}{z_0^2}\left(\left\|\mathbf{r}_1\right\|^2 - \left\|\mathbf{r}_2\right\|^2\right) = 0 \\ \left\|\alpha\mathbf{v}_1 + \mathbf{v}_2\right\|^2 - \left\|\beta\mathbf{v}_1 + \mathbf{v}_3\right\|^2 \\ \qquad = \alpha^2 - \beta^2 + \left\|\mathbf{v}_2\right\|^2 - \left\|\mathbf{v}_3\right\|^2\end{cases} \\ \rightarrow \alpha^2 - \beta^2 + \left\|\mathbf{v}_2\right\|^2 - \left\|\mathbf{v}_3\right\|^2 = 0.\end{aligned} \tag{17}$$

From Eq. (16), we have

$$\beta = -\frac{\mathbf{v}_2^\mathsf{T}\mathbf{v}_3}{\alpha}. \tag{18}$$

Plugging this into Eq. (17) results in a bi-quadratic (fourth-degree) equation in α, *i.e.*,

$$\alpha^4 + \left(\|\mathbf{v}_2\|^2 - \|\mathbf{v}_3\|^2\right)\alpha^2 - \left(\mathbf{v}_2^\mathsf{T}\mathbf{v}_3\right)^2 = 0. \tag{19}$$

Bi-quadratic equations have at most four real solutions that can be solved by using the quadratic formula. Complex number calculations are avoidable by simply checking the sign of the discriminant, allowing us to compute real solutions very easily and stably. Once the real solution(s) for α is obtained, substituting α into Eq. (18) yields β.

Finally, we can recover the rotation matrix $\mathtt{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^\mathsf{T}$ by

$$\begin{aligned} \mathbf{r}_1 &= \frac{1}{\|\alpha\mathbf{v}_1 + \mathbf{v}_2\|}(\alpha\mathbf{v}_1 + \mathbf{v}_2), \\ \mathbf{r}_2 &= \frac{1}{\|\beta\mathbf{v}_1 + \mathbf{v}_3\|}(\beta\mathbf{v}_1 + \mathbf{v}_3), \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2, \end{aligned} \tag{20}$$

the depth to the reference plane by

$$z_0 = \frac{\mathbf{r}_1^\mathsf{T}\mathbf{a}_1 + \mathbf{r}_2^\mathsf{T}\mathbf{a}_2}{2}, \tag{21}$$

and the translation vector by

$$\mathbf{t} = z_0 \begin{bmatrix} \mathbf{m}_g \\ 1 \end{bmatrix} - \mathtt{R}\mathbf{X}_g. \tag{22}$$

3.2. Para perspective

In the para perspective case, the procedure is actually identical to the weak perspective case up to the computation of the nullspace vector. Specifically, subtract the centroids from the image points and corresponding 3D points, respectively, *i.e.*, $\hat{\mathbf{m}}_i^p = \mathbf{m}_i^p - \mathbf{m}_g$, $\hat{\mathbf{X}}_i = \mathbf{X}_i - \mathbf{X}_g$, and define

$$\mathbf{p} = \frac{1}{z_0}\left(\mathbf{r}_1 - x_0\mathbf{r}_3\right), \quad \mathbf{q} = \frac{1}{z_0}\left(\mathbf{r}_2 - y_0\mathbf{r}_3\right), \tag{23}$$

which yields

$$\begin{bmatrix} \hat{\mathbf{m}}_1^p & \hat{\mathbf{m}}_2^p \end{bmatrix} = \begin{bmatrix} \mathbf{p}^\mathsf{T} \\ \mathbf{q}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}_1 & \hat{\mathbf{X}}_2 \end{bmatrix}. \tag{24}$$

Repeating the steps from Eq. (7) through Eq. (15) then produces the three nullspace vectors $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$ in the same manner as in the weak perspective case.

The approach for computing α and β differs from the weak perspective case. For the vectors $\mathbf{p}$ and $\mathbf{q}$, we can construct the following three equations:

$$\begin{aligned} \|\mathbf{p}\|^2 &= \begin{cases} \dfrac{\|\mathbf{r}_1 - x_0\mathbf{r}_3\|^2}{z_0^2} = \dfrac{1 + x_0^2}{z_0^2} \\ \|\alpha\mathbf{v}_1 + \mathbf{v}_2\|^2 = \alpha^2 + \|\mathbf{v}_2\|^2 \end{cases} \\ &\rightarrow \frac{1}{z_0^2} = \frac{\|\mathbf{v}_2\|^2 + \alpha^2}{1 + x_0^2}, \end{aligned} \tag{25}$$

$$\begin{aligned} \|\mathbf{q}\|^2 &= \begin{cases} \dfrac{\|\mathbf{r}_2 - y_0\mathbf{r}_3\|^2}{z_0^2} = \dfrac{1 + y_0^2}{z_0^2} \\ \|\beta\mathbf{v}_1 + \mathbf{v}_3\|^2 = \beta^2 + \|\mathbf{v}_3\|^2 \end{cases} \\ &\rightarrow \frac{1}{z_0^2} = \frac{\|\mathbf{v}_3\|^2 + \beta^2}{1 + y_0^2}, \end{aligned} \tag{26}$$

$$\begin{aligned} \mathbf{p}^\mathsf{T}\mathbf{q} &= \begin{cases} \dfrac{(\mathbf{r}_1 - x_0\mathbf{r}_3)^\mathsf{T}(\mathbf{r}_2 - y_0\mathbf{r}_3)}{z_0^2} = \dfrac{x_0 y_0}{z_0^2} \\ (\alpha\mathbf{v}_1 + \mathbf{v}_2)^\mathsf{T}(\beta\mathbf{v}_1 + \mathbf{v}_3) = \alpha\beta + \mathbf{v}_2^\mathsf{T}\mathbf{v}_3 \end{cases} \\ &\rightarrow \frac{1}{z_0^2} = \frac{\alpha\beta + \mathbf{v}_2^\mathsf{T}\mathbf{v}_3}{x_0 y_0}. \end{aligned} \tag{27}$$

Note that $(x_0, y_0) = (u_g, v_g)$. We eliminate z_0^2 by selecting two equation pairs such as Eqs. (25) and (27) and Eqs. (25) and (26), then obtain two polynomials in α and β, *i.e.*,

$$\begin{aligned} (x_0 y_0)\|\mathbf{p}\|^2 - (1 + x_0^2)\mathbf{p}^\mathsf{T}\mathbf{q} \\ = k_1\alpha^2 + k_2\alpha\beta + k_3 = 0, \end{aligned} \tag{28}$$

$$\begin{aligned} (1 + y_0^2)\|\mathbf{p}\|^2 - (1 + x_0^2)\|\mathbf{q}\|^2 \\ = k_4\alpha^2 + k_5\beta^2 + k_6 = 0, \end{aligned} \tag{29}$$

where

$$\begin{aligned} k_1 &= -x_0 y_0, \\ k_2 &= x_0^2 + 1, \\ k_3 &= \mathbf{v}_2^\mathsf{T}\mathbf{v}_3(x_0^2 + 1) - \|\mathbf{v}_2\|^2 x_0 y_0, \\ k_4 &= y_0^2 + 1, \\ k_5 &= -(x_0^2 + 1), \\ k_6 &= \|\mathbf{v}_2\|^2(y_0^2 + 1) - \|\mathbf{v}_3\|^2(x_0^2 + 1). \end{aligned} \tag{30}$$

From Eq. (28), we have

$$\beta = -\frac{k_1\alpha^2 + k_3}{k_2\alpha}. \tag{31}$$

Now we can obtain a bi-quadratic equation in α by substituting Eq. (31) into Eq. (29):

$$\left(k_1^2 k_5 + k_2^2 k_4\right)\alpha^4 + \left(2k_1 k_3 k_5 + k_2^2 k_6\right)\alpha^2 + k_3^2 k_5 = 0. \tag{32}$$

As in the weak perspective case, the real solution(s) of α is obtainable by using the quadratic formula, and β is also determined by Eq. (31).

Once we obtained α and β, we can calculate $\mathbf{p} = \alpha\mathbf{v}_1 + \mathbf{v}_2$ and $\mathbf{q} = \alpha\mathbf{v}_1 + \mathbf{v}_3$. To recover the motion parameters, we first find the constant depth

$$z_0 = \frac{1}{2}\left(\frac{\sqrt{1 + x_0^2}}{\|\mathbf{p}\|} + \frac{\sqrt{1 + y_0^2}}{\|\mathbf{q}\|}\right), \tag{33}$$

for which we utilized Eqs. (25) and (26).

Then, we determine the rotation matrix $\mathtt{R}$. By rearranging Eq. (23), we obtain

$$\mathbf{r}_1 = z_0\mathbf{p} + x_0\mathbf{r}_3, \quad \mathbf{r}_2 = z_0\mathbf{q} + y_0\mathbf{r}_3. \tag{34}$$

Due to the constraint $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$, we have

$$\begin{aligned}\mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ &= (z_0\mathbf{p} + x_0\mathbf{r}_3) \times (z_0\mathbf{q} + y_0\mathbf{r}_3) \\ &= z_0^2\,(\mathbf{p}\times\mathbf{q}) + y_0 z_0\,(\mathbf{p}\times\mathbf{r}_3) - x_0 z_0\,(\mathbf{q}\times\mathbf{r}_3)\,.\end{aligned} \tag{35}$$

Thus, we can derive $\mathbf{r}_3$ by

$$\mathbf{r}_3 = z_0^2\left(\mathtt{I} - y_0 z_0[\mathbf{p}]_\times + x_0 z_0[\mathbf{q}]_\times\right)^{-1}(\mathbf{p}\times\mathbf{q})\,, \tag{36}$$

where $[\mathbf{p}]_\times$ denotes a 3×3 skew symmetric matrix representing the vector cross product such that $\mathbf{x}\times\mathbf{y} = [\mathbf{x}]_\times\mathbf{y}$. By substituting $\mathbf{r}_3$ into Eq. (34), we obtain $\mathbf{r}_1$ and $\mathbf{r}_2$. Finally, we can determine $\mathbf{t}$ by Eq. (22) as in the weak perspective case.

3.3. From affine to perspective

The rotation matrix $\mathtt{R}$ and translation vector $\mathbf{t}$ calculated in Secs. 3.1 and 3.2 are based on an affine camera model; consequently, applying them to standard perspective cameras may yield low accuracy. To address this issue, we upgrade the affine P3P solution to the exact P3P solution. We adopt a P3P formulation by Ke *et al*. [24] and optimize the rotation matrix $\mathtt{R}$ so that the constraints are satisfied.

We first outline Ke *et al*. 's formulation. From Eq. (1), using two pairs of 2D–3D correspondences we can eliminate the translation vector as follows

$$z_i^c\begin{bmatrix}\mathbf{m}_i\\1\end{bmatrix} - z_j^c\begin{bmatrix}\mathbf{m}_j\\1\end{bmatrix} = \mathtt{R}\,(\mathbf{X}_i - \mathbf{X}_j)\,. \tag{37}$$

We remove the depths z_i^c, z_j^c by taking the vector cross product of the image points:

$$\left(\begin{bmatrix}\mathbf{m}_i\\1\end{bmatrix}\times\begin{bmatrix}\mathbf{m}_j\\1\end{bmatrix}\right)^\mathsf{T}\mathtt{R}\,(\mathbf{X}_i - \mathbf{X}_j) = \mathbf{c}_{ij}^\mathsf{T}\,\mathtt{R}\mathbf{d}_{ij} = 0. \tag{38}$$

Accordingly, starting from the rotation obtained by the weak/para P3P solver, we refine it so as to satisfy Eq. (38).

We apply Newton's method on the Lie algebra with a small-angle approximation [23]. Approximating the incremental rotation as $\Delta\mathtt{R} \approx \mathtt{I} + [\Delta\mathbf{r}]_\times$, where $\Delta\mathbf{r}$ is a 3×1 rotation vector, the constraint equation becomes

$$\mathbf{c}_{ij}^\mathsf{T}\left(\mathtt{I} + [\Delta\mathbf{r}]_\times\right)\mathtt{R}\mathbf{d}_{ij} = 0. \tag{39}$$

We can obtain the increment $\Delta\mathbf{r}$ by solving the resulting linear system

$$\begin{bmatrix}(\mathbf{c}_{12}\times\mathtt{R}\mathbf{d}_{12})^\mathsf{T}\\(\mathbf{c}_{23}\times\mathtt{R}\mathbf{d}_{23})^\mathsf{T}\\(\mathbf{c}_{31}\times\mathtt{R}\mathbf{d}_{31})^\mathsf{T}\end{bmatrix}\Delta\mathbf{r} = \begin{bmatrix}\mathbf{c}_{12}^\mathsf{T}\,\mathtt{R}\mathbf{d}_{12}\\\mathbf{c}_{23}^\mathsf{T}\,\mathtt{R}\mathbf{d}_{23}\\\mathbf{c}_{31}^\mathsf{T}\,\mathtt{R}\mathbf{d}_{31}\end{bmatrix}. \tag{40}$$

The exact incremental rotation is computed by the exponential-mapping or Rodrigues' formula:

$$\begin{aligned}\Delta\mathtt{R} &= \exp([\Delta\mathbf{r}]_\times) \\ &= \mathtt{I} + \sin\theta\left[\frac{\Delta\mathbf{r}}{\theta}\right]_\times + (1-\cos\theta)\left[\frac{\Delta\mathbf{r}}{\theta}\right]_\times^2,\end{aligned} \tag{41}$$

where $\theta = \|\Delta\mathbf{r}\|$. Then, we upgrade the rotation via

$$\mathtt{R} \leftarrow \Delta\mathtt{R}\,\mathtt{R}. \tag{42}$$

Iterating from Eq. (39) to Eq. (42) until sufficiently small $\Delta\mathbf{r}$, we can calculate the rotation matrix $\mathtt{R}$ that coincides with the solution to the exact P3P problem.

After obtained $\mathtt{R}$, from Eq. (1), eliminating the depth via the vector cross product leads to a linear system in $\mathbf{t}$:

$$\begin{bmatrix}[\mathbf{m}_1]_\times\\[\mathbf{m}_2]_\times\end{bmatrix}\mathbf{t} = -\begin{bmatrix}[\mathbf{m}_1]_\times\mathtt{R}\mathbf{X}_1\\[\mathbf{m}_2]_\times\mathtt{R}\mathbf{X}_2\end{bmatrix}. \tag{43}$$

We can obtain $\mathbf{t}$ by solving the above linear equations.

There are two advantages to using the above iterative scheme. First, the per-iteration computational cost for solving the linear system is low compared to a joint optimization of both rotation and translation. Minimization over $[\Delta\mathbf{r}, \mathbf{t}] \in \mathbb{R}^6$ is a commonly used approach for camera pose tracking in Visual SLAM [25, 31]. However, the linear system is of size 6×6 whereas Eq. (40) is at most 3×3. Second, a valid rotation is guaranteed at any iteration. In distance-based P3P formulations that minimize pairwise distances [12, 14, 33, 36], the distant constraints need to be exactly zero to obtain a valid rotation, which prevents early termination of the refinement. In contrast, our upgrade scheme ensures that the rotation remains on SO(3) even if the optimization stops at any iterations.

4. Experiments

This section reports evaluation experiments conducted using both synthetic and real-image datasets. The primary objective is to investigate how closely the performance of the affine solvers approach that of exact P3P solvers. Accordingly, we compared variants of our proposed methods only with the Ding method [12], which is the current state-of-the-art and outperforms other solvers such as [24, 26, 33, 36].

Specifically, we evaluated the weak P3P solver described in Sec. 3.1, the para P3P solver in Sec. 3.2, and their variants incorporating the upgrade procedure detailed in Sec. 3.3. We indicate the number of upgrade iterations by a subscript in the method name: for instance, Weak_0 refers to the vanilla weak P3P solver without any upgrade, and Para_2 denotes the para P3P solver with two upgrade iterations.

We also integrated each solver with the vanilla RANSAC [13] and LO-RANSAC [10]. To focus on a fair comparison with the Ding method, we did not employ more

recent RANSAC variants such as GC-RANSAC [4]. The reprojection error threshold was set to 3 pixels, and we configured RANSAC to achieve a success probability of 0.995. EPnP [28] implemented in OpenCV was adopted as the local optimizer. After convergence, we applied the reprojection error minimization by Levenberg-Marquardt method, specifically OpenCV's solvePnP with `ITERATIVE` option, to the predicted inliers to further refine the camera pose estimation. The initial maximum number of iterations was adjusted depending on the type of experiment, and we updated the iteration upper-bound every time more inliers are found. To draw same samples in different minimal solvers, we fixed the random seed for each trial.

We calculated the rotation errors by the rotation angle $\epsilon_{\mathtt{R}} = \mathrm{acos}((\mathrm{trace}(\mathtt{R}_{\mathrm{gt}}^{\mathsf{T}}\mathtt{R}_{\mathrm{est}}) - 1)/2)$ [degrees] and the translation errors by the relative L2 distance $\epsilon_{\mathbf{t}} = \min(\|\mathbf{t}_{\mathrm{gt}} - \mathbf{t}_{\mathrm{est}}\| / \|\mathbf{t}_{\mathrm{gt}}\|, 1.0) \times 100$ [%].

All experiments were implemented in MATLAB. For the Ding method, we ported the official C++ code[1] to MATLAB. Evaluations were conducted on a PC equipped with an Intel Core i7-13700 processor.

4.1. Synthetic data evaluation

We generated simulation data as follows. The camera was configured with a 1024×1024 resolution, principal point at (512, 512), and a $45°$ field of view (approximately a 1236-pixel focal length). We sampled N random image points in pixel coordinates and normalized them using the inverse intrinsics to obtain $\mathbf{m}_i = (u_i, v_i)$ in the normalized image plane. Next, we drew a random camera-frame depth z_i^c and formed 3D point $\mathbf{X}_i = z_i^c[u_i, v_i, 1]^{\mathsf{T}}$, establishing N 2D–3D correspondences $\{\mathbf{m}_i \leftrightarrow \mathbf{X}_i \mid i \in \{1, \ldots, N\}\}$. Finally, we sampled a random camera pose, $\mathtt{R}$ and $\mathbf{t}$, and applied the rigid transformation $\mathbf{X}_i \leftarrow \mathtt{R}^{\mathsf{T}}(\mathbf{X}_i - \mathbf{t})$.

4.1.1. Depth deviation sensitivity

We first investigated the sensitivity of the proposed affine P3P solvers to depth variation. When generating $N = 3$ correspondences, we set the depths as $z_1^c = z_0$, $z_2^c = (1 + \Delta z/z_0)z_0$, and $z_3^c = (1 - \Delta z/z_0)z_0$, and then reprojected onto the normalized image plane to obtain the 2D measurements $\mathbf{m}_{1,2,3}$. No image noise was added.

Figure 2 shows the median rotation, translation, and reprojection errors over 10^6 independent trials for each Δz, with $0 \leq |\Delta z/z_0| \leq 0.5$. Ding's solver produced zero errors irrespective of depth variation because it solves P3P problem exactly. The vanilla affine solvers, Weak_0 and Para_0, are sensitive to depth variation; Para_0 is slightly more accurate than Weak_0 due to its higher approximation, but both are fragile. In contrast, the iterative upgrade dramatically improves accuracy. Two correction steps (Weak_2 and Para_2) converged to an exact solution on par with Ding. We observed similar results for 2D points near the image boundary, which are shown in the supplementary materials.

[1] https://github.com/yaqding/P3P/blob/main/matlab/solver_p3p.cpp

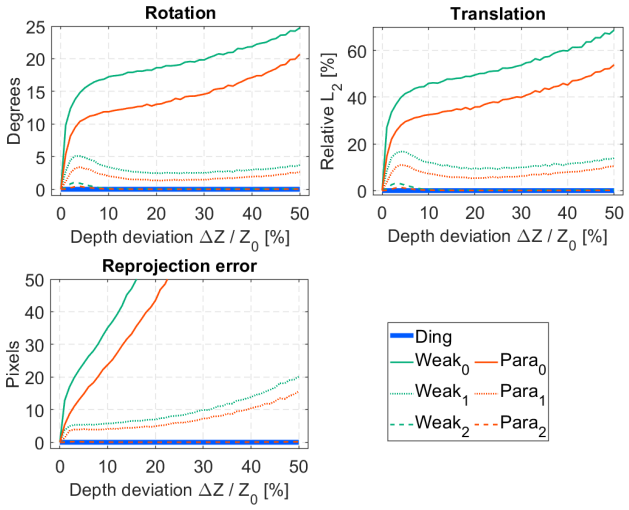

Figure 2. Sensitivity against depth deviation. Median values over 10^6 independent trials for each depth. Weak_2 and Para_2 almost overlap with Ding.

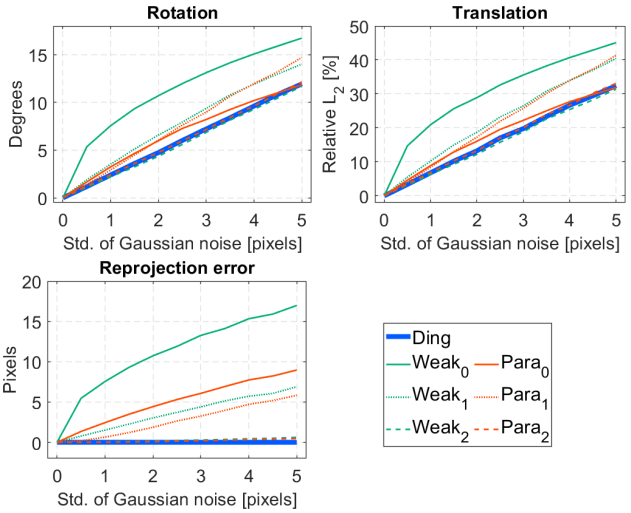

Figure 3. Robustness against image noise. Median values over 10^6 independent trials for each noise level. Weak_2 and Para_2 almost overlap with Ding.

4.1.2. Image noise robustness

We next evaluated robustness to image noise. To remove the influence of depth variation, we fixed the depth of all three 3D points to $z_i^c = z_0$, *i.e.*, $\Delta z = 0$ for all 3D points. We then added zero-mean Gaussian noise with standard deviation σ to the image measurements in pixels.

Figure 3 reports the median rotation, translation, and reprojection errors over 10^6 independent trials for each σ,

varying $0 \leq \sigma \leq 5$ pixels. Even without depth variation, Weak_0 and Para_0 are vulnerable to image noise. As in the depth sensitivity test, the upgrade procedure is effective: even starting from affine's low-accuracy initialization, two upgrade steps achieves performance comparable to Ding.

4.1.3. Computational time

With the noise robustness test, we concurrently measured runtime of the solvers: 50.1 µsec by Ding, 40.3 µsec by Weak_0, 44.2 µsec by Weak_1, 45.5 µsec by Weak_2, 42.0 µsec by Para_0, 46.7 µsec by Para_1, 48.1 µsec by Para_2. The proposed affine P3P solvers are more efficient than Ding because they rely solely on bi-quadratic equations. With one upgrade step Weak_1 and Para_1 are approximately 10% faster than Ding; with two steps Weak_2 and Para_2 are still approximately 4% faster.

4.1.4. RANSAC scenario

We evaluated each solver as a minimal solver within RANSAC when correspondences are contaminated by outliers. We considered two depth-variations: small variation with $|\Delta z/z_0| \leq 0.1$ and large variation with $-0.5 \leq \Delta z/z_0 \leq 1.0$. We generated $N = 1000$ correspondences and replaced a portion of them with random outliers, varying the outlier ratio from 0.1 to 0.8. We added Gaussian image noise with $\sigma = 1$ pixel to the inlier 2D points. Given the success probability of 0.995 and a minimal sample size of 3, the theoretical maximum number of iterations at 80% outliers is $\log(1-0.995)/\log(1-0.2^3) \approx 660$; in the experiment we set 1000 iterations as the upper-bound.

For each outlier ratio, we ran 1000 independent trials and evaluated RANSAC and LO-RANSAC instantiated with the proposed solvers and with Ding. Figure 4 summarizes the results of median values. The affine approximation held reasonably well for small depth deviations, where both Weak_0 and Para_0 converge even with noticeable estimation errors. Under large depth variation, their convergence rate drops sharply as the outlier ratio increases. Their average runtime seems decreasing as the outlier ratio grew; this does not indicate speedup—it simply reflected more frequent failures to converge, which in turn avoids the cost of the final Levenberg-Marquardt refinement. Consistent with Secs. 4.1.1 and 4.1.2, two upgrade steps (Weak_2 and Para_2) delivered performance indistinguishable from Ding.

4.2. Real data evaluation

We report experimental evaluations on public real-image datasets shown in Fig. 5. The synthetic results in Sec. 4.1 suggested that the vanilla affine P3P solvers (Weak_0 and Para_0) are not accurate enough in some situations, but can still be useful within LO-RANSAC when depth deviations are small. They also showed that the upgraded variants (Weak_2 and Para_2) reaches the accuracy of the Ding's solver irrespective of depth variation. Based on these findings, in

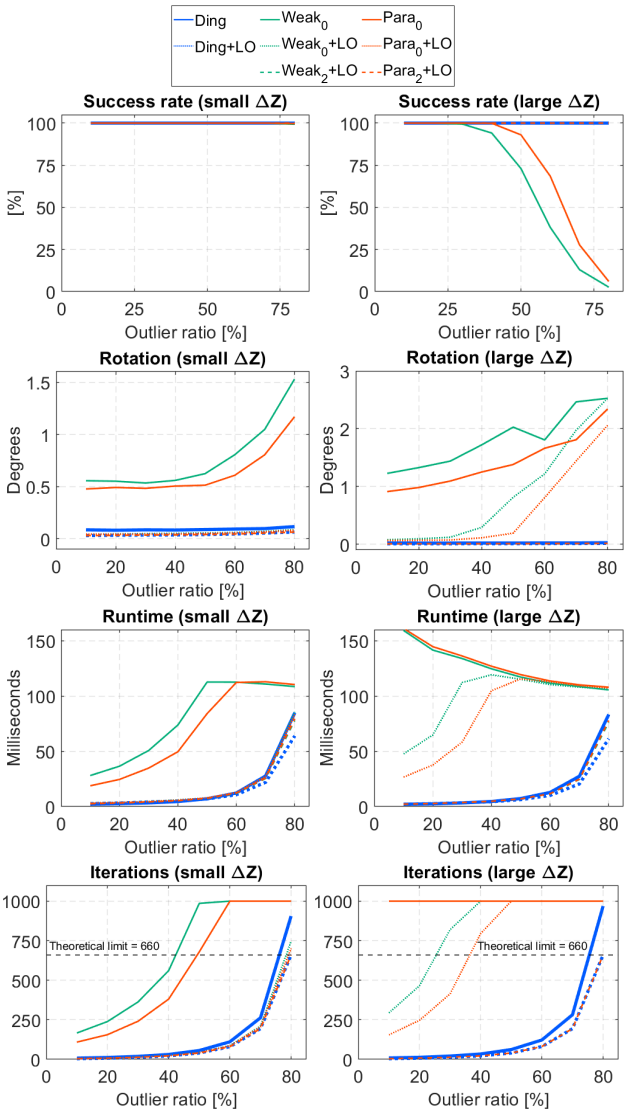

Figure 4. RANSAC simulation. Median over 10^3 trials for each outlier ratio. Weak_2 and Para_2 almost overlap with Ding.

this experiment, we instantiated LO-RANSAC with Weak_0, Para_0, Weak_2, Para_2, and Ding, and compared them on the real datasets. We set the inlier threshold to 3 pixels and limited the maximum number of LO-RANSAC iterations at 2000. To ensure fair random sampling, we used the same random seed across different solvers for the same input.

4.2.1. 6D object pose estimation

We first evaluated under small depth deviation. The EPOS dataset [20, 30] is a subset of the BOP Challenge 2019 for 6D object pose estimation [19], comprising three sequences: T-LESS, YCB-V, and LM-O. As shown in the first column of Fig. 5, a single image contains multiple objects, and for each object the 2D points concentrate in a portion of the image. For each object in a image, the dataset pro-

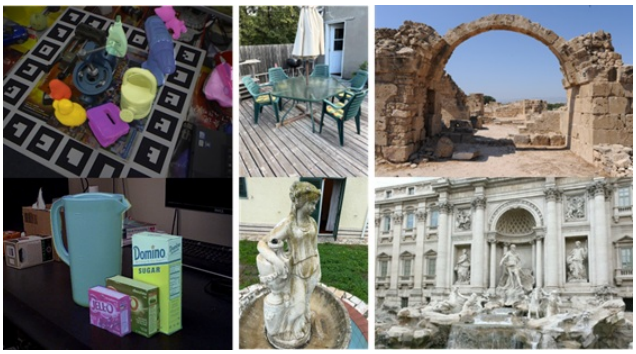

Figure 5. Real image dataset. *Left*: the EPOS dataset [20, 30]. *Middle and Right*: the IMC2023 dataset [9].

vides 2D–3D correspondences that include outliers as well as the ground-truth 6D pose of the object. The total number of instances is approximately 6,600.

Table 1 reports the recall value (the fraction of instances whose error is below threshold) together with the total number of iterations and the total processing time elapsed by LO-RANSAC to process all instances. Consistent with the synthetic data experiments, Weak_2 and Para_2 exhibit nearly identical accuracy to Ding, with no clear difference in processing time and total iterations. Since object depths are relatively small with respect to the distance from the camera, Weak_0 and Para_0 perform better than in the synthetic evaluations. Para_0 shows slightly higher recall than Weak_0, which empirically supports its theoretically higher approximation. Per-sequence results and recall w.r.t. runtime are provided in the supplementary material.

4.2.2. Camera localization

We next evaluated under large depth deviation using the IMC 2023 dataset [9], which is an outdoor image dataset for Structure-from-Motion benchmarking. It consists of four categories (Haiper, Heritage, Phototourism, Urban) and contains roughly 5,800 images in total. As shown in the second and third columns of Fig. 5, scenes span a wide range of distances from near to far. Unlike EPOS, 2D points are distributed across the entire image. not in a small region.

Each image sequence stores 2D-3D point correspondences and the camera motion, which were reconstructed by COLMAP [38]. Additionally, each sequence contains outlier 2D points, which were detected but not successfully associated with any 3D point. To simulate outlier 2D-3D pairs in an image, we synthesized outlier 3D points sampled from a normal distribution whose mean and standard deviation were calculated from the inlier 3D points visible in the image.

Table 2 summarizes the recall values, the total number of LO-RANSAC iterations, and total runtime. As in the previous experiments in Secs. 4.1 and 4.2.1, Weak_2 and Para_2 demonstrate comparable performance to Ding. Since many sequences have substantial depth variation, both Weak_0 and Para_0 are inferior to Ding. In such cases, the difference in approximation accuracy between Weak_0 and Para_0 does not appear to produce a practical difference. Results by category and frequency of the real solutions are shown in the supplementary material.

Table 1. Quantitative results on the EPOS dataset. Recalls in percentages, and runtime and RANSAC iterations elapsed to process all data.

Method	Recall (ϵ_R / ϵ_t) ↑			Time↓	Iters↓
	2°/2%	3°/3%	5°/5%	(min.)	($\times 10^6$)
Ding +LO	8.3	19.3	36.8	14.1	2.32
Weak_2+LO	8.4	19.3	37.1	14.6	2.32
Para_2 +LO	8.5	19.3	37.1	14.7	2.32
Weak_0+LO	7.7	17.7	36.5	11.9	2.41
Para_0 +LO	7.9	18.9	37.0	13.4	2.33

Table 2. Quantitative results on the IMC2023 dataset. Recalls in percentages, and runtime and RANSAC iterations elapsed to process all data.

Method	Recall (ϵ_R / ϵ_t) ↑			Time↓	Iters↓
	0.5°/1%	1°/2%	3°/3%	(min.)	($\times 10^6$)
Ding +LO	25.1	40.9	51.9	38.0	1.51
Weak_2+LO	25.0	40.9	51.9	39.2	1.52
Para_2 +LO	25.1	40.7	51.7	38.9	1.52
Weak_0+LO	23.5	39.8	50.7	44.6	2.00
Para_0 +LO	23.4	39.8	50.7	44.3	1.91

5. Conclusions

We have derived direct P3P solutions to the weak perspective and para perspective camera models and have evaluated their practicality through extensive experiments. Since the affine camera models are merely an approximation to perspective projection, we have shown that the vanilla affine P3P solvers are useful, limited to scenes with small depth deviation. Yet, starting with the affine approximations and performing a few upgrade iterations yields accuracy equivalent to the SOTA exact P3P solver. Our findings suggest that, beyond revisiting the P3P problem, more complex camera-geometry problems—P4Pfr, P1P, multi-view geometry *etc*.—may also be approached effectively by leveraging affine cameras. Although not addressed due to the lack of public datasets, evaluations of our solvers to super-zoom/telecentric cameras is raised as future work.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 1

[2] T.D. Alter. 3-d pose from 3 points using weak-perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):802–808, 1994. 1, 2

[3] Pei An, Jiaqi Yang, Muyao Peng, You Yang, Qiong Liu, Xiaolin Wu, and Liangliang Nan. Mincd-pnp: Learning 2d-3d correspondences with approximate blind pnp. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 26519–26528, 2025. 1

[4] Daniel Barath and Jiří Matas. Graph-cut ransac. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6733–6741, 2018. 6

[5] Ronen Basri. Paraperspective≡ affine. *International Journal of Computer Vision*, 19(2):169–179, 1996. 2

[6] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 1

[7] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *Asian Conference on Computer Vision*, pages 11–24. Springer, 2010. 1

[8] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[9] Ashley Chow, Eduard Trulls, HCL-Jevster, Kwang Moo Yi, lcmrll, old ufo, Sohier Dane, tanjigou, WastedCode, and Weiwei Sun. Image matching challenge 2023, 2023. 8

[10] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *Joint pattern recognition symposium*, pages 236–243. Springer, 2003. 1, 5

[11] Daniel F DeMenthon and Larry S Davis. Model-based object pose in 25 lines of code. In *European conference on computer vision*, pages 335–343. Springer, 1992. 1

[12] 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, 5

[13] 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, 5

[14] Xiaoshan Gao and Hangfei Chen. New algorithms for the perspective-three-point problem. *Journal of Computer Science and Technology*, 16(3):194–207, 2001. 5

[15] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 1

[16] Johann August Grunert. Das pothenotische problem in erweiterter gestalt nebst bber seine anwendungen in der geodasie. *Grunerts Archiv fur Mathematik und Physik*, pages 238–248, 1841. 1

[17] Levente Hajder. W-pnp method: optimal solution for the weak-perspective n-point problem and its application to structure from motion. In *International Conference on Computer Vision Theory and Applications*, pages 265–276. SciTePress, 2017. 1

[18] 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(3):331–356, 1994. 1

[19] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2019. 7

[20] Tomáš Hodaň, Dániel Baráth, and Jiří Matas. EPOS: Estimating 6D pose of objects with symmetries. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7, 8

[21] Radu Horaud, Fadi Dornaika, and Bart Lamiroy. Object pose: The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22(2):173–189, 1997. 2

[22] T.S. Huang, A.M. Bruckstein, R.J. Holt, and A.N. Netravali. Uniqueness of 3d pose under weak perspective: a geometrical proof. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1220–1221, 1995. 2

[23] Kenichi Kanatani. *3D Rotations: Parameter Computation and Lie Algebra based Optimization*. Chapman and Hall/CRC, 2020. 5

[24] 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. 1, 5

[25] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 5

[26] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011. 1, 5

[27] Viktor Larsson, Torsten Sattler, Zuzana Kukelova, and Marc Pollefeys. Revisiting radial distortion absolute pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1062–1071, 2019. 1

[28] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009. 6

[29] C-P Lu, Gregory D Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE transactions on pattern analysis and machine intelligence*, 22(6):610–622, 2000. 1

[30] Jiri Matas, Rene Ranftl, Daniel Barath, Ondra Chum, Tat-Jun Chin, and Dmytro Mishkin. Ransac in 2020: A cvpr tutorial, 2020. 7, 8

[31] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 5

[32] Gaku Nakano. A versatile approach for solving pnp, pnpf, and pnpfr problems. In *European Conference on Computer Vision*, pages 338–352. Springer, 2016. 1

[33] Gaku Nakano. A simple direct solution to the perspective-three-point problem. In *BMVC*, page 26, 2019. 1, 5

[34] Denis Oberkampf, Daniel F DeMenthon, and Larry S Davis. Iterative pose estimation using coplanar feature points. *Computer Vision and Image Understanding*, 63(3):495–511, 1996. 1

[35] Yu-ichi Ohta, Kiyoshi Maenobu, and Toshiyuki Sakai. Obtaining surface orientation from texels under perspective projection. In *IJCAI*, pages 746–751, 1981. 1, 2

[36] 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. 1, 5

[37] Conrad J Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE transactions on pattern analysis and machine intelligence*, 19(3):206–218, 1997. 1

[38] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 8

[39] Carsten Steger. Algorithms for the orthographic-n-point problem. *Journal of Mathematical Imaging and Vision*, 60 (2):246–266, 2018. 1

[40] George Terzakis and Manolis Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. In *European Conference on Computer Vision*, pages 478–494. Springer, 2020. 1

[41] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154, 1992. 1

[42] 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

[43] Jonathan Ventura, Zuzana Kukelova, Torsten Sattler, and Dániel Baráth. Absolute pose from one or two scaled and oriented features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20870–20880, 2024. 1

[44] Changchang Wu. P3.5p: Pose estimation with unknown focal length. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2440–2448, 2015. 1

[45] Guangyang Zeng, Shiyu Chen, Biqiang Mu, Guodong Shi, and Junfeng Wu. Cpnp: Consistent pose estimator for perspective-n-point problem with bias elimination. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1940–1946. IEEE, 2023. 1

[46] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Astrom, and Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2344–2351, 2013. 1