sQPEP: Global Optimal Solutions to Scaled Quadratic Pose Estimation Problems

Bohuan Xue, Yilong Zhu, Tianyu Liu, Jin Wu, Jianhao Jiao, Yi Jiang,
Chengxi Zhang, Xinyu Jiang and Zhijian He

***Abstract*—State estimation encounters significant hurdles in scale ambiguity, both when assimilating data from scale-uninformed sources such as Structure from Motion (SfM) and when handling normalized point clouds, each scenario demanding robust solutions to achieve consistent scale and accurate estimation. Addressing this critical issue, we propose the Scaled Quadratic Pose Estimation Problem (sQPEP), a novel unified framework designed to enhance scale estimation in various state estimation algorithms. Our framework not only establishes a globally optimal solution strategy for the precise estimation of pose and scale factors but also systematically categorizes a broad spectrum of pose estimation challenges. This is crucial for advancing our theoretical understanding and the practical application of these solutions. The sQPEP framework consolidates a range of scale and pose estimation challenges into a unified theoretical paradigm, thereby refining the methodology for these estimations. By applying algebraic techniques, we have effectively bifurcated the problem into two distinct categories. Subsequently, we have deduced globally optimal solutions and unveiled two robust solvers. These solvers are proficient in generating 80 and 81 solutions for their respective problem classes, featuring elimination template dimensions of 664×744 and 521×602. Our method's efficacy has been rigorously confirmed through experimental validation, which demonstrates its consistent performance in degenerate conditions and its superior noise immunity. These results bolster the framework's applicability to intricate scenarios encountered in real-world settings.**

***Index Terms*—calibration, pose estimation, Gröbner basis, polynomial**

I. Introduction

INDUSTRIAL robotics employs pose estimation to enable precise identification of position and orientation across various applications [1]–[4]. In efforts to estimate pose, robotic systems commonly utilize a suite of sensors, including cameras, LiDAR, and IMUs. These sensors are pivotal in developing models for perspective projection, geometric alignment, and spatial registration to enable accurate spatial localization and orientation detection. Common pose estimation problems include the Perspective-n-Point, hand-eye calibration problem, point-to-point, point-to-plane, and more, with numerous specialized algorithms for similar problems, such as [5]–[8]. Subsequently, [9] formulated the optimization objectives as quadratic terms of pose elements, thereby unifying them into a Quadratic Pose Estimation Problem (QPEP) framework.

Accurate scale estimation is crucial in various applications. In robotics, it affects the precision of a robotic arm's interactions with objects during hand-eye calibration. In 3D reconstruction and AR/VR, it determines the alignment between virtual models and physical space, impacting user experiences. Autonomous vehicles and medical imaging also rely on precise scale factors for safe navigation and accurate diagnoses, respectively.

However, various solutions assume known scale information, i.e., 6DOF pose estimation. The importance of accurate scale estimation is often overlooked in these solutions. Nevertheless, there exist cases of 7DOF estimation where the scale is unknown. This challenge is particularly evident in the context of robotics and computer vision, where two of the most common active sensors, cameras and LiDAR, often face situations where effective scale information is missing.

Monocular cameras are inherently unable to provide scale information [10], leading to the need for additional reference objects [11], [12] or multiple sensors [13]–[16] to recover scale. However, most of these methods are highly specialized and tailored to specific tasks, which makes it challenging to generalize them to other application scenarios. LiDAR sensors, on the other hand, face scale drift due to factors such as temperature and pressure variations, which affect the laser's wavelength and emission power [17]. Ignoring these scale changes often leads to suboptimal point cloud registration even with ground truth poses.

In light of the limitations of existing methods and the importance of accurate scale estimation, we propose the Scaled Quadratic Pose Estimation Problem (sQPEP) framework. Unlike previous approaches that either assume known scale or rely on additional information, our framework directly

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. (*Corresponding author: Zhijian He*)

B. Xue is with the School of Data Science and Engineering, and Xingzhi College, South China Normal University. Shanwei, 516622, China. Email: bxue@ieee.org

Y. Zhu, T. Liu and J. Wu are with the Hong Kong University of Science and Technology, Hong Kong SAR, China. Email:{yzhubr,tliubk}@connect.ust.hk, jin_wu_uestc@hotmail.com

J. Jiao is with the Department of Computer Science, University College London. Email: jiaojh1994@gmail.com

Y. Jiang is with the City University of Hong Kong, Hong Kong SAR, China. Email: JY369356904@163.com

C. Zhang is with School of Internet of Things, Jiangnan University, Wuxi, 214122, China. Email dongfangxy@163.com

X. Jiang is with Department of Computer and Information Science, University of Macau, Macau 999078, China. Email yc27912@um.edu.mo

Z. He is with Shenzhen Technology University, Shenzhen, 518118, China. Email: hezhijian@sztu.edu.cn

addresses the 7DOF pose estimation problem, jointly optimizing the pose and scale. This unified approach not only tackles a broader range of challenges but also provides a more principled and efficient solution for scale-aware pose estimation. Detailed derivations and the solvers are available in the supplementary material at: `https://github.com/byronsit/sQPEP_Solver`.

A. *Challenges*

Despite the central role of pose estimation in advancing intelligent systems, achieving a universally applicable solution is hindered by several major challenges. Among these, the diverse range of problem formulations is perhaps most prominent. Custom-tailored to specific scenarios and scales, these formulations have generated a fragmented landscape of solutions, lacking the generalizability required for widespread application. This fragmentation is exacerbated by the broad solution spaces that emerge in the absence of prior knowledge about scale factors, making the search for optimal solutions increasingly complex.

Moreover, the numerical stability of algorithms becomes a significant challenge when the scale variable is introduced. The optimization process may amplify numerical errors due to the scale factor, leading to instability in the final pose estimation. Coupled with this is the issue of scale coupling, where the scale factor is intertwined with rotation and translation. As a result, adjustments to the scale factor can impact the optimal rotation and translation, meaning that scale cannot always be precomputed as with point matching problems [18].

B. *Contributions*

In the fields of computer vision and robotics, many tasks are commonly defined in general terms without explicit residual formulations. Different types of residuals require distinct optimization schemes, and the globally optimal computational methods vary accordingly.

Heller et al. [19] utilize the inter-frame image registration relationships and ϵ-epipolar constraints to transform the original optimization objective into minimizing $||e||_\infty$. They finally employ Linear Programming to determine the bounding step of the algorithm. Wu et al. [20] engage in complex symbolic mathematical derivations, converting the original problem into a sum of residuals involving rotation and translation: $\sum \operatorname{tr}(\boldsymbol{E}_i^\top \boldsymbol{E}_i) + \boldsymbol{v}_i^\top \boldsymbol{v}$, which results in a polynomial of the highest order of 15, and solve this complex polynomial using the tool Mathematica. The residual form for the hand-eye calibration problem as described in QPEP [9] directly utilizes the Frobenius norm defined by $||\boldsymbol{AX} - \boldsymbol{XB}||$, thus posing the problem as a quadratic pose estimation problem. However, QPEP do not provide a detailed derivation process, requiring people to transform the problem case-by-case into a system of polynomial equations that satisfy specific properties and solve them using Groebner bases.

Our approach differs fundamentally from the aforementioned methods, particularly in our residual equation which incorporates an additional scaling factor, thus altering the residual formulation. Our method extends the content of QPEP and bears similarities only with QPEP; other methods show little resemblance. For hand-eye calibration problem, We also use the Frobenius norm to define our residual as $||\boldsymbol{A}(\boldsymbol{s})\boldsymbol{X} - \boldsymbol{XB}||$. By interpreting the scale s differently in various contexts, we categorize sQPEP into two types and provide specific mathematical forms, enabling any reader to directly verify if their problem falls under sQPEP, as detailed in Supplementary Material. Our method allows us to derive a general solver for all sQPEP problems, eliminating the need, unlike QPEP, to derive solutions separately for different tasks. Unlike previous methods that optimize over 6DOF, our approach employs a 7DOF optimizer, a distinction that we wish to emphasize.

The development of a comprehensive framework capable of overcoming these obstacles is crucial. Such a framework would enable standardized comparisons across various methods and adapt to different scales and conditions, thereby markedly enhancing the robustness and adaptability of pose estimation methodologies.

In this work, we introduce a generic framework for addressing the scaled Quadratic Pose Estimation Problem (sQPEP), a challenging conundrum in many field. Our approach bifurcates the problem into two distinct formulations. Utilizing mathematical ingenuity, we delve into the nuances of each form, eventually devising a globally optimal solution through the application of Gröbner basis techniques.

This paper makes three main contributions:

- **Mathematical Formulation:** We introduce a definitive mathematical formulation of QPEP and sQPEP, establishing a framework for new pose estimation problems.
- **Algebraic Solver Development:** By dissecting sQPEP's variants, we utilized algebraic methods and Gröbner bases to develop two distinct solvers for sQPEP variations.
- **Experimental Validation:** Extensive testing confirms our method's robustness under degeneracy and noise, with the added benefits of numerical stability and real-time performance on limited memory, validating our approach's efficiency.

C. *Outline*

The structure of this paper is organized as follows: Section II formulates the problem and introduces the solutions we propose. Section III details the experiments we have conducted to evaluate our proposed solutions. Section IV summarizes our results and suggests future research paths.

II. Problem Statement and Solutions

A. *Notations*

In this section, we define the notations used within our study: s represents the scale factor to be estimated. $\boldsymbol{q} = [q_x, q_y, q_z, q_w]^\top$ denotes the quaternion of the pose that we aim to estimate. $\boldsymbol{t} = [t_x, t_y, t_z]^\top$ is the translation vector. $\boldsymbol{t_s} = [t_x, t_y, t_z, s]^\top$, which combines the translation vector with the scale factor. vec: The vec operator, as introduced by Neudecker, arranges the elements of a matrix into a vector. $\otimes$: The Kronecker product. $\boldsymbol{V}^p_{a\times b}$ is a constant matrix with dimensions $a \times b$, and $\boldsymbol{V}_a$ is a constant vector of dimension $a \times 1$. The superscript p is used to distinguish constant matrices of the same dimensions. p may be omitted when the context is clear. Each $\boldsymbol{V}$ represents a unique constant matrix or vector in this paper. C stands for a constant term, each C in this paper represents a unique constant. $\boldsymbol{q}^d$ represents the vector formed by the elements of $\boldsymbol{q}$ under the d-Veronese mapping.

B. Problem Statement and Derived Definition

As delineated in [9], the quadratic pose estimation problem is expressed as the optimization task:

$$\arg\min_{\boldsymbol{q},\boldsymbol{t}} \mathcal{L}(\boldsymbol{q}, \boldsymbol{t}), \quad \text{s.t.} \quad \boldsymbol{q}^\top \boldsymbol{q} = 1, \boldsymbol{t} \in \mathbb{R}^3,$$

where the objective function $\mathcal{L}$ is not explicitly defined. To advance our discussion and provide a precise formulation of the function $\mathcal{L}$, we leverage the property delineated in [9]: the sum of two QPEP problems remains a QPEP. By aggregating all problems possessing the QPEP property, we arrive at the following equation:

$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{t}) = \boldsymbol{V}_{35}^\top \boldsymbol{q}^4 + \boldsymbol{V}_{40}^\top \text{vec}(\boldsymbol{q}^2 \otimes \tilde{\boldsymbol{t}}) + \boldsymbol{V}_6{}^\top \boldsymbol{t}^2 + \boldsymbol{V}_3{}^\top \boldsymbol{t} + C,$$

in which $\tilde{\boldsymbol{t}}$ is the homogeneous translation such that $\tilde{\boldsymbol{t}} = \left(\boldsymbol{t}^\top, 1\right)^\top$. Based on the established framework, we are able to articulate a definitive characterization of the scaled quadratic pose estimation problem.

Similarly, we seek to establish a unified formalization of the sQPEP problem for 7DOF pose estimation. Depending on the placement of the scale factor, sQPEP can be divided into two distinct categories: the first category involves the scale factor applied to the translation component:

$$\begin{aligned}\mathcal{L}_{s1}(\boldsymbol{q}, \boldsymbol{t}, s) = {} & \boldsymbol{V}_{35}^\top \boldsymbol{q}^4 + \boldsymbol{V}_{30}^\top \text{vec}(\boldsymbol{q}^2 \otimes \boldsymbol{t}) + s \cdot \boldsymbol{V}_{10}^\top \boldsymbol{q}^2 + \\ & \boldsymbol{V}_6{}^\top \boldsymbol{t}^2 + s \cdot \boldsymbol{V}_4{}^\top \boldsymbol{t_s} + \boldsymbol{V}_4{}^\top \boldsymbol{t_s} + C\end{aligned}$$

The second category involves the scale factor applied to the rotation component:

$$\mathcal{L}_{s2}(\boldsymbol{q}, \boldsymbol{t}, s) = s^2 \cdot \boldsymbol{V}_{35}^\top \boldsymbol{q}^4 + s \cdot \boldsymbol{V}_{40}^\top \text{vec}(\boldsymbol{q}^2 \otimes \tilde{\boldsymbol{t}}) + \boldsymbol{V}_6{}^\top \boldsymbol{t}^2 + \boldsymbol{V}_3{}^\top \boldsymbol{t} + C.$$

Depending on the characteristics of various problems, some elements of the constant matrix $\boldsymbol{V}$ may be zero. Typically, this simplifies the problem, so we will focus our discussion on the most complex scenario where none of the elements of $\boldsymbol{V}$ are zero. For a detailed derivation of the aforementioned formula, refer to Appendix A.

C. Solution Strategies for $\mathcal{L}_{s_1}$

The method of Lagrange multipliers offers a powerful solution for optimization problems with complex constraints, where traditional methods may struggle. By introducing Lagrange multipliers, this technique transforms a constrained problem into an unconstrained one, simplifying the solution process.

With this in mind, we opt to employ the method of Lagrange multipliers to study the optimization of the function $\mathcal{L}_{s_1}$. To commence, we define an auxiliary Lagrangian function $\mathcal{X}_1$ that incorporates not only the original objective function $\mathcal{L}_{s_1}(\boldsymbol{q}, \boldsymbol{t}, s)$ but also a term associated with the constraint $\boldsymbol{q}^\top \boldsymbol{q} = 1$ involving a Lagrange multiplier. Hence, our Lagrangian function can be expressed as:

$$\mathcal{X}_1 = \mathcal{L}_{s_1}(\boldsymbol{q}, \boldsymbol{t}, s) - 1/2 \cdot \lambda(\boldsymbol{q}^\top \boldsymbol{q} - 1) \tag{1}$$

Note that we introduce $\frac{1}{2}\lambda$ as the coefficient for the multiplier term to simplify the derivation process in subsequent calculations. Next, we will seek the solution to the optimization problem by setting the partial derivatives of $\mathcal{X}_1$ with respect to $\boldsymbol{q}$ and λ to zero.

First, we consider the derivatives with respect to the translation:

$$\begin{cases} \frac{\partial \mathcal{X}_1}{\partial t_i} = \boldsymbol{V}_{4\times1}^{i\top} \boldsymbol{t_s} + \boldsymbol{V}_{10\times1}^{i\top} \boldsymbol{q}^2 + C_{t_i} \\ \frac{\partial \mathcal{X}_1}{\partial s} = \boldsymbol{V}_{4\times1}^{\top} \boldsymbol{t_s} + \boldsymbol{V}_{10\times1}^{\top} \boldsymbol{q}^2 + C_s, \end{cases} \tag{2}$$

where $t_i \in \{t_x, t_y, t_z\}$. By combining (2) we can derive $\boldsymbol{t_s} = \boldsymbol{V}_{4\times10}^\top \boldsymbol{q}^2$. It is observed that $\frac{\partial \mathcal{X}_1}{\partial q_i} = \boldsymbol{V}_{20\times1}^\top \boldsymbol{q}^3 + \boldsymbol{V}_{4\times1}^\top vec(\boldsymbol{q} \otimes s) + \boldsymbol{V}_{12\times1}\text{vec}(\boldsymbol{q} \otimes \boldsymbol{t}) - \lambda q_i$, where q_i denotes the quaternion components. Combining with $\boldsymbol{t_s}$ and after simplifying we can get $\frac{\partial \mathcal{X}_1}{\partial q_i} = \boldsymbol{V}_{20\times1}^{\text{x}\top} \boldsymbol{q}^3 - \lambda q_{\text{i}}$.

Lemma 1: Let $f(x_1, x_2, \ldots, x_m) = C\prod_{i=1}^m x_i^{p_i}$ be a polynomial function in m variables, where C is a constant and p_i are non-negative integers, and the sum $\sum_{i=1}^m p_i = n$ represents the total degree of the polynomial, Let $S = x_1, x_2, \ldots, x_m$. Define $\boldsymbol{D}^{(n)}$ as the vector of all possible nth-order partial derivatives of f, where the partial derivatives are arranged lexicographically, consistent with the arrangement of $\text{vec}(S^{\otimes n})$. Then, the product of the nth-order partial derivatives of f with respect to all its variables and $\text{vec}(S^{\otimes n})$ satisfies the following relationship:

$$n! \cdot f = \boldsymbol{D}^{(n)} \cdot \text{vec}(S^{\otimes n}) \tag{3}$$

The explicit form of $\boldsymbol{D}^{(n)}$ is given by:

$$\boldsymbol{D}^{(n)} = \left[\frac{\partial^n f}{\partial x_1^{k_1} \partial x_2^{k_2} \ldots \partial x_m^{k_m}} \right]_{n=\sum_{i=1}^m k_i,\, k_i \geq 0}.$$

The lemma 1 illustrates a method of reinterpreting the dot product as matrix multiplication, which, when combined (3) with $\frac{\partial \mathcal{X}_1}{\partial q_i}$ and incorporating the additional elements of $\boldsymbol{q}$, yields the equation:

$$\boldsymbol{W}_{4\times64} \cdot \text{vec}(\boldsymbol{q}^{\otimes3})_{64\times1} = \lambda \boldsymbol{q}. \tag{4}$$

$\boldsymbol{W}$ is assumed to be a constant matrix. (4) and $\boldsymbol{q}^\top \boldsymbol{q} = 1$ are polynomial equations for which a Gröbner basis is an effective tool for finding solutions [21]. The use of Gröbner bases to solve systems of multivariate polynomials is a classical approach, details of which can be found in [22] and [21]. Briefly, the process begins by computing the Gröbner basis of the system, which consists of a set of polynomials that generate the same ideal as the original system. Subsequently, a set of monomials is selected as a basis, and the expression of each basis element under the action of each polynomial in the Gröbner basis is computed. This results in a multiplication matrix that describes the multiplication relations among the basis elements. Next, each polynomial in the original system is simplified and expressed as a linear combination of the basis elements. If the multiplication matrix is perfect, this linear combination should precisely replicate the original polynomial; however, typically, some error terms arise. These error terms are represented as a linear combination of the original equations, and the coefficients form an elimination matrix. This matrix yields a new set of equations that hold within the ideal generated by the original system. Adding these new equations to the original system results in an elimination template, which facilitates the simplification and acceleration of subsequent computations. This solver features an elimination template with dimensions 664×744 and provides 80 distinct solutions. By evaluating the residuals of all real solutions in $\mathcal{L}_{s1}$, the global optimum for this category of problems can be determined. For more details, refer to our Supplementary Material, which provides specific examples of how to use the elimination template and extends the methodology to the experimental section that follows.

D. Solution Strategies for $\mathcal{L}_{s_2}$

Similar to the formulation defined by $\mathcal{L}_{s1}$ in (1), we can establish

$$\mathcal{X}_2 = \mathcal{L}_{s_2}(\boldsymbol{q}, \boldsymbol{t}, s) - 1/2\lambda(\boldsymbol{q}^\top \boldsymbol{q} - 1). \tag{5}$$

Taking the derivative with respect to the translation components yields: $\frac{\partial \mathcal{X}_2}{\partial t_i} = \boldsymbol{V}_{3\times1}^{i\top} \boldsymbol{t} + \boldsymbol{V}_{10\times1}^{i\top} \cdot \boldsymbol{q}^2 \cdot s + C_i$. However, when differentiating with respect to s, it emerges that $\frac{\partial \mathcal{X}_2}{\partial s} = \sum_{j\in\{x,y,z\}} \boldsymbol{V}_{10\times1}^{j\top} \cdot \boldsymbol{q}^2 \cdot t_j + \boldsymbol{V}_{35\times1}^{\top} \cdot \boldsymbol{q}^4 \cdot s$. This implies that it is no longer possible to form equations in the style of (2) and $\boldsymbol{t_s}$, due to the coupling of s with $\boldsymbol{q}$. We can express $\boldsymbol{t}$ as $\boldsymbol{t} = \boldsymbol{V}_{3\times4} \cdot \boldsymbol{q}^2 \cdot s + \boldsymbol{V}_{3\times1}$. Upon substituting the expression from $\boldsymbol{t}$ into $\frac{\partial \mathcal{X}_2}{\partial s}$, we obtain the following system of partial derivatives:

$$\begin{cases} \frac{\partial \mathcal{X}_2}{\partial s} = \boldsymbol{V}_{10}^{\top} \cdot \boldsymbol{q}^2 + \boldsymbol{V}_{35}^{\top} \cdot \boldsymbol{q}^4 \cdot s \\ \frac{\partial \mathcal{X}_2}{\partial q_i} = \boldsymbol{V}_{4}^{\top} \cdot \boldsymbol{q} \cdot s + \boldsymbol{V}_{20}^{\top} \cdot \boldsymbol{q}^3 \cdot s^2 - \lambda q_i \end{cases} \tag{6}$$

However, employing the Gröbner basis to the system of equations (6) results in an ideal with a Krull dimension of 2, denoting the solution space as two-dimensional with an infinite number of solutions. Traditional solvers, which are adept at handling finite, discrete sets of solutions, are hence inadequate for directly deriving all solutions.

Subsequently, We intend to reformulate the expression of (6) to continue leveraging the Gröbner basis approach for solution determination, and to devise a solver capable of generating specific solution forms.

By examining $\boldsymbol{t}$ and (6), we find that s is coupled exclusively with $\boldsymbol{q}^2$, allowing us to express the quaternion as $\tilde{\boldsymbol{q}} = \sqrt{s}\boldsymbol{q}$ using the following equation to represent the quaternion, which enables us to reformulate $\mathcal{X}_2$ as

$$\tilde{\mathcal{X}}_2(\tilde{\boldsymbol{q}}, \boldsymbol{t}) = \boldsymbol{V}_{35}^{\top} \tilde{\boldsymbol{q}}^4 + \boldsymbol{V}_{40}^{\top} \mathrm{vec}(\tilde{\boldsymbol{q}}^2 \otimes \begin{bmatrix} \boldsymbol{t} \\ 1 \end{bmatrix}) + \boldsymbol{V}_{6}^{\top} \boldsymbol{t}^2 + \boldsymbol{V}_{3}^{\top} \boldsymbol{t} + C. \tag{7}$$

It is worth noting that unlike (5), $\tilde{\mathcal{X}}_2$ no longer requires the Lagrange multiplier, λ, to constrain $\boldsymbol{q}$, significantly reducing the complexity of the problem.

Upon differentiating (7) with respect to the translation components, we obtain $\frac{\partial \tilde{\mathcal{X}}_2}{\partial t_i} = \boldsymbol{V}_{3}^{i\top} \cdot \boldsymbol{t} + \boldsymbol{V}_{11}^{i\top} \cdot \tilde{\boldsymbol{q}}^2 + C_i$. Through straightforward algebraic computation, we can deduce the expression for $\boldsymbol{t}$ as $\boldsymbol{t} = \boldsymbol{V}_{3\times11} \cdot \mathrm{vec}(\tilde{\boldsymbol{q}}^2 \otimes [\boldsymbol{t}, 1]^\top)$. By substituting it into (7) and differentiating with respect to $\tilde{\boldsymbol{q}}$, we derive $\frac{\partial \tilde{\mathcal{X}}_2}{\partial \tilde{q}_i} = \boldsymbol{V}_{4}^{i\top} \cdot \tilde{\boldsymbol{q}} + \boldsymbol{V}_{20}^{i\top} \cdot \tilde{\boldsymbol{q}}^3$,. By combining it with (3), we can obtain

$$\tilde{\boldsymbol{W}}_{4\times64} \cdot \mathrm{vec}(\tilde{\boldsymbol{q}}^{\otimes3})_{64\times1} = \boldsymbol{Q}_{4\times4}\tilde{\boldsymbol{q}} \tag{8}$$

Here, the constant matrices $\tilde{\boldsymbol{W}}$ and $\boldsymbol{Q}$ can be directly derived. (8) is also a polynomial equation which, similarly to (4), can be solved using a Gröbner basis. This yields a solver with an elimination template of size 521×602 that has 81 solutions. With this, the problem of sQPEP is addressed. In the following chapter, we will empirically evaluate our approach through a series of experiments designed to test its efficacy and robustness in various scenarios.

Algorithm 1 Optimization Algorithm for sQPEP

```
Require: The initial optimization objective function.
Ensure: The globally optimal solution.
Construct the optimization objective function, and check if it is a $s$QPEP which can be written as either $\mathcal{L}_{s1}$ or $\mathcal{L}_{s2}$.
if it can be written as $\mathcal{L}_{s1}$ then
    Obtain matrices $W$ according to (1)-(4).
else if it can be written as $\mathcal{L}_{s2}$ then
    Obtain $\tilde{W}$ and $Q$ according to (5) and (7)-(8).
end if
Obtain all local minima of the problem by using a Solver.
if it is written as $\mathcal{L}_{s1}$ then
    Solve using the elimination template of size $664 \times 744$ generated by (4).
else if it is written as $\mathcal{L}_{s2}$ then
    Solve using the elimination template of size $521 \times 602$ generated by (8).
end if
Substitute all real solutions from the solver into the original residual equations, and determine the globally optimal solution by the smallest residual, discarding any invalid solutions based on prior knowledge, such as the sign of $S$
```

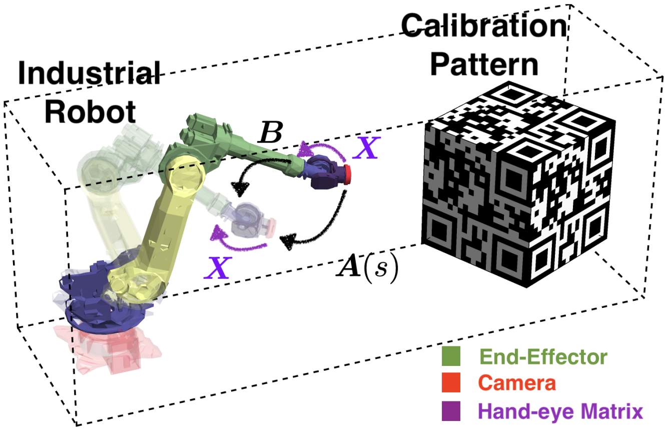


Fig. 1. What is hand-eye calibration? In this context, $\boldsymbol{A}(s)$ or $\boldsymbol{A}$ represents the camera's displacement in the world, typically expressed using a transformation matrix. $\boldsymbol{B}$ denotes the displacement of the Robot End-Effector in the world, also expressed using a transformation matrix. $\boldsymbol{X}$ defines the pose relationship between the camera and the Robot End-Effector, which is a constant and the target of our estimation. $\boldsymbol{A}(s)$ represents the camera's motion trajectory, with an unknown scale factor. If the calibration pattern does not provide effective scale information (such as edge lengths), the scale s must be estimated. This implies that in such cases, $\boldsymbol{A}(s)$ is used to express the camera's motion trajectory. For a more detailed explanation, see Supplement B.

III. Experimental Results

A. Overview

In this section, we demonstrate the applicability of our proposed framework through two specific applications: hand-

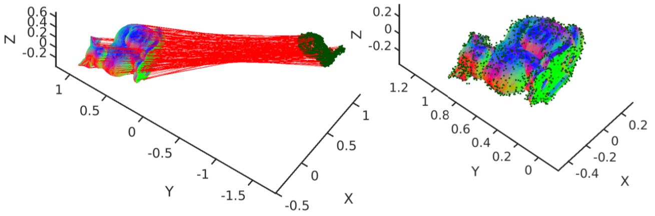


Fig. 2. What is Point-to-Plane? Left: Original state of point cloud and associated surfaces. The **red lines** indicate the correspondences between matched points, while the **multi-colored surfaces** denote the directions of different surface normals. The **dark green** points represent the points that are to be matched. Right: Resultant configuration after computation with our solver, showcasing the alignment of point correspondences and the orientation of surface normals. We aim to estimate the pose relationship between the dark green and the multicolored rabbits, characterized by rotation $\boldsymbol{R}$, translation $\boldsymbol{t}$, and scale factor s. By enlarging the point cloud of the original dark green rabbit, followed by rotation and translation, alignment with the multicolored rabbit is achieved. The image on the right is an example where the points of the dark green rabbit closely match the multicolored rabbit.

eye calibration and point-to-plane registration, both of which are crucial for accurate robotic perception, manipulation and computer vision. We compare the performance of our method across various applications. Detailed explanations on how these two applications are formulated as sQPEP and solved using our approach are provided in the Supplementary Material. The algorithmic procedures are summarized in Algorithm 1.

Fig. 1 demonstrates the hand-eye calibration process. In this context, we formulate an optimization problem to find the transformation matrix $\boldsymbol{X}$ that minimizes the residual errors between the robotic hand (manipulator) and the eye (camera). The objective function is defined as:

$$\arg\min_{\boldsymbol{q},\boldsymbol{t},s} \sum_{i=1}^{\mathcal{N}} \operatorname{tr}\left((\boldsymbol{A_i X} - \boldsymbol{X B_i})^\top (\boldsymbol{A_i X} - \boldsymbol{X B_i})\right),$$

where $\boldsymbol{X}$ denotes the sought transformation matrix. with $\boldsymbol{X}$ representing the transformation matrix between the robotic hand (manipulator) and the eye (camera). The matrix $\boldsymbol{X}$ is given by: $\boldsymbol{X} = \begin{pmatrix} \boldsymbol{R}(\boldsymbol{q}) & \boldsymbol{t} \\ \boldsymbol{0}_{3\times1} & 1 \end{pmatrix}$ where $\boldsymbol{q}$ is transformed into a rotation matrix [23]. The parameters to estimate include $\boldsymbol{q}$, $\boldsymbol{t}$, and s is the scale with the translation in $\boldsymbol{A}_i$. Each $\boldsymbol{A}_i$ is a homogeneous transformation matrix represented as: $\boldsymbol{A}_i = \begin{pmatrix} \boldsymbol{R}_{ai} & s\cdot\boldsymbol{t}_{ai} \\ \boldsymbol{0}_{3\times1} & 1 \end{pmatrix}$ and $\boldsymbol{B}_i$ is another homogeneous transformation matrix: $\boldsymbol{B}_i = \begin{pmatrix} \boldsymbol{R}_{bi} & \boldsymbol{t}_{bi} \\ \boldsymbol{0}_{3\times1} & 1 \end{pmatrix}$ This setup constitutes a first-class sQPEP problem where all constant vectors in $\mathcal{L}_{s1}$ are non-zero. For a more detailed explanation of hand-eye calibration and how it is transformed into $\mathcal{L}_{s1}$, please refer to our supplementary material.

Fig. 2 illustrates the point-to-plane registration problem. This registration approach aims to minimize the squared perpendicular distances between a set of points and planes. The optimization problem is formulated as:

$$\arg\min_{\boldsymbol{q},\boldsymbol{t},s} \sum_{i=1}^{\mathcal{N}} \left(\boldsymbol{n}_i^\top (s\boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{t} - \boldsymbol{y}_i)\right)^2,$$

where $\boldsymbol{n}_i$ represents the normal vector to the plane at the i-th point, $\boldsymbol{R}$ and $\boldsymbol{t}$ denote the rotation and translation components of the transformation, respectively, and s is a scaling factor. $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are the corresponding points in the dataset and on the plane, respectively. This is recognized as a second-class sQPEP with no zero constants in $\mathcal{L}_{s2}$.

The proposed algorithms were implemented in MATLAB 2023a on a computer with an i7 8700K CPU and 42GB of RAM, ensuring robust computational capability for the experiments. To objectively evaluate the performance of our system, we defined three error metrics:

- Rotation Error: $\epsilon_r = \arccos\left(\frac{1}{2}(\operatorname{tr}(\boldsymbol{R}^\top \boldsymbol{R}_{gt}) - 1)\right)$, which quantifies the angular difference between the estimated and actual rotations.
- Translation Error: $\epsilon_t = \|\boldsymbol{t} - \boldsymbol{t}_{gt}\|_2$, which measures the Euclidean distance error in translation estimations.
- Scale Error: $\epsilon_s = \|s - s_{gt}\|_2$, which assesses the variance in scale estimation relative to the ground truth.

μ_{ϵ_t} and μ_{ϵ_r} denote the mean translation and rotation errors, while $\sigma^2_{\epsilon_t}$ and $\sigma^2_{\epsilon_r}$ represent their variances across multiple experiments, providing a comprehensive evaluation of the algorithm's performance and consistency.

B. Experiments on Simulated Environments for $\mathcal{L}_{s_1}$

In industrial scenarios, it is often necessary to dynamically calibrate cameras with other motion sensors, such as extrinsic calibration between cameras and LiDAR, wheel odometry, or GNSS, as well as extrinsic calibration of odometry generated by LiDAR with partially inaccurate scales and other sensors. In these situations, it is typically impossible to provide markers with known scales. These problems fall under the category of hand-eye calibration without scale, which requires estimating the 7DOF transformation between sensors. Unfortunately, since the seminal work of [10], no new viable 7DOF estimation algorithms have emerged. As a result, in practical applications, we are usually limited to comparing with other 6DOF estimation algorithms.

The dataset created for this investigation included a comprehensive array of transformations, employing two distinct scaling factors, specifically 0.5 and 2.5, to cater to a broad spectrum of motion magnitudes. These transformations were subjected to Gaussian noise with a mean of zero and standard deviations of 0.01, 0.05, and 0.1 to approximate real-world measurement inaccuracies. The noise was infused directly into the scaling and translational parameters, whereas rotational disturbances were imparted to the 3×3 rotation matrices within $SO(3)$, with subsequent normalization to maintain the orthogonality property of $SO(3)$. To ensure statistical significance and robustness of the results, each parameter configuration was tested over 100 independent trials. The method [10] was selected as the baseline for our comparative work. Additionally, for comparative analysis, the Gauss Newton (GN) and Quasi-Newton (QN) methods were employed, utilizing GT rotation and scale values as initial conditions, with the translation vector set to zero. The convergence precision was set to 10^{-12}, and a maximum of 1000 iterations was allowed.

In the preliminary experiment, translational components of each pose were sampled from a Gaussian distribution with zero mean and a standard deviation of one, while the rotational

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3540135

TABLE I
HAND-EYE CALIBRATION RESULTS ON SIMULATED DATA WITH NOISE=0.01 AND SCALE=0.5

	μ_{ϵ_r} (deg)	$\sigma^2_{\epsilon_r}$ (deg^2)	μ_{ϵ_t} (m)	$\sigma^2_{\epsilon_t}$ (m)^2
ours	0.3308	0.2259	0.0223	0.0180
Andreff [10]	0.3081	**0.2072**	**0.0221**	**0.0177**
Gauss Newton	8.9753	9.2618	1.0937	0.7594
Quasi Newton	13.8109	11.1328	0.7509	0.5427
Liang [24]	**0.2152**	0.1264	1.5833	0.6802
Wu-4D [25]	0.3062	0.1679	1.6836	0.6952
QPEP [9]	2.8512	2.8546	1.5935	0.6852
Sarabandi [26]	1.9122	8.9833	1.5831	0.6797

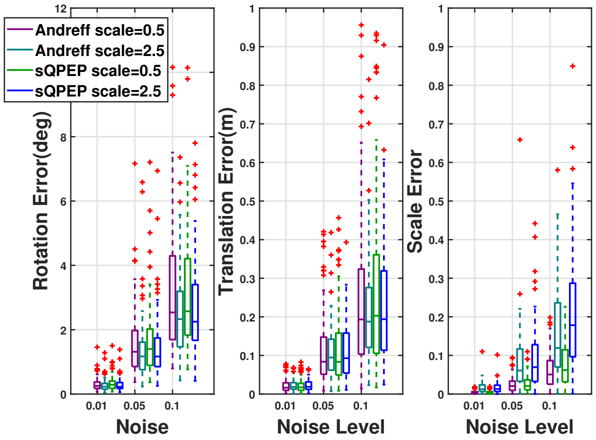


Fig. 3. $\mathcal{L}_{s_1}$ Experimental Results on a well-conditioned dataset for hand-eye calibration

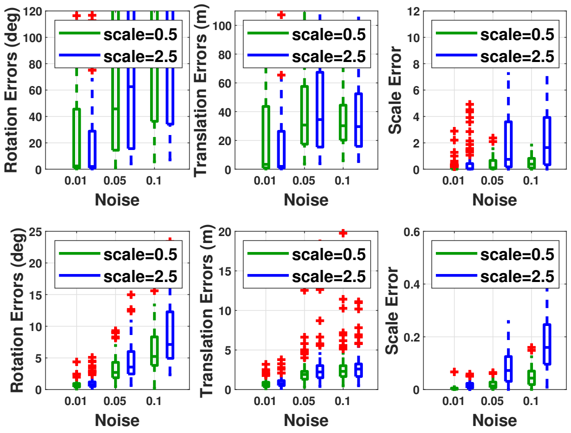


Fig. 4. $\mathcal{L}_{s_1}$ Experimental Results on an ill-conditioned dataset for hand-eye calibration. *Top*: Method by Andreff et al. [10], *Bottom*: Proposed sQPEP method.

components were uniformly distributed over the $SO(3)$ space. This approach ensured a dataset with a broad and robust range of motion, devoid of singularities and exhibiting sufficient excitation in both rotational and translational aspects. The performance of various algorithms is presented in Table I. All 6DOF method fail to obtain reliable translation data, which also implies an inability to correctly recover the scale. Therefore, we focus more on the performance differences compared to [10] in the subsequent analysis. The synthesized dataset's effectiveness in mirroring real calibration conditions is illustrated in Fig. 3. Due to differences in the objective functions, [10] achieved slightly better performance on the well-conditioned dataset. However, subsequent experiments revealed limitations in their algorithm. The GN and QN may still converge to incorrect local optima even when initiated at almost the true value positions. A detailed analysis of this phenomenon is provided in the supplementary material.

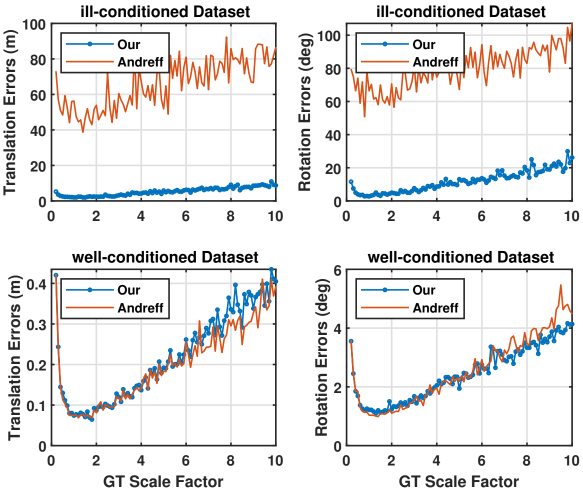


Fig. 5. $\mathcal{L}_{s_1}$ Experimental : Performance comparisons between our method and Andreff's method under various scales with noise level 0.05. The upper figure shows an ill-conditioned dataset where Andreff's method completely fails, while the lower figure shows a well-conditioned dataset where both methods exhibit distinct advantages.

Next, we introduced a variation in the dataset where each rotation was based on the identity quaternion, with disturbances in the quaternion parameters having a standard deviation of 0.01, followed by normalization to the $SO(3)$ space. Under these ill-conditioned conditions, the comparative performance, as shown in Fig. 4. In light of the fact that Andreff's method does not achieve global optimality, it consistently yields erroneous solutions. In practice, the proximity of multiple local optima renders conventional techniques ineffective at distinguishing the accurate solution. In contrast, our global optimization approach, sQPEP, demonstrates a substantial advantage when confronted with these types of challenges.

We next conduct comparisons on well-conditioned at various scales, with results presented in Fig. 5 and Fig. 6. Increasing or decreasing the scale parameter effectively magnifies the difference between two factors, thereby introducing errors. Since the datasets are constructed based on a zero-mean Gaussian distribution, when the scale parameter is set to one, translation and rotation impacts are equivalently scaled, enabling all algorithms to achieve optimal performance. As the scale parameter increases, our method and the [10] show different strengths in handling rotations and translations, respectively. This variation is attributed to different optimization goals and the unequal inherent weights assigned to translation and rotation across algorithms. For a more detailed discussion, readers are referred to [27] for additional insights.

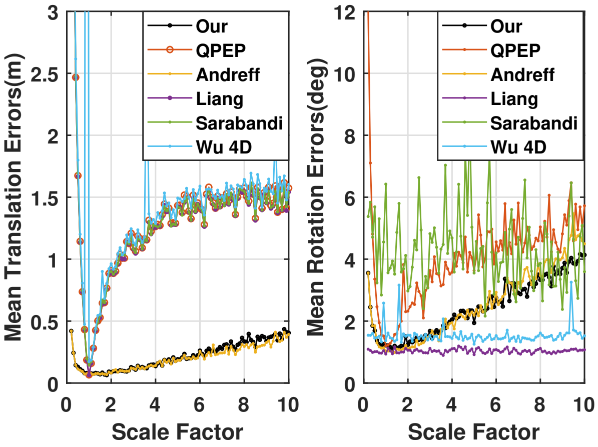


Fig. 6. $\mathcal{L}_{s_1}$ Experimental : Performance comparison of all hand-eye calibration algorithms on well-conditioned datasets across different scales with

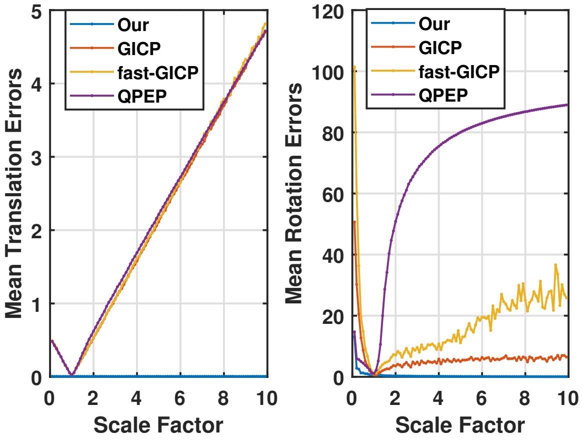


Fig. 7. $\mathcal{L}_{s_2}$ Experimental : Performance comparison of different algorithms at varying scales with noise level 0.05. Note that our algorithm is represented in **blue** and closely follows the x-axis.

TABLE II
RESULT OF SCALED POINT-TO-PLANE PROBLEM WITH NOISE=0.01 AND SCALE=0.5

	μ_{ϵ_r}(deg)	$\sigma^2_{\epsilon_r}$(deg^2)	μ_{ϵ_t}(m)	$\sigma^2_{\epsilon_t}$(m^2)
ours	0.2339	0.0979	0.0011	0.0004
GICP [8](GT)	7.4063	4.9358	0.2678	0.0111
Fast-GICP [28](GT)	6.8399	4.4320	0.2673	0.0109
QPEP [9]	4.3824	0.0985	0.2719	0.0053
GICP [8](Random)	10.6648	22.9984	0.2698	0.0196
Fast-GICP [28](Random)	12.8037	28.9876	0.2713	0.0249

C. Experiments on Simulated Environments for $\mathcal{L}_{s_2}$

Sensors such as LiDAR, which can acquire depth data, might be utilized for scene localization purposes. In cases where the device scale is imprecise, it is essential to conduct 7DOF pose estimation that incorporates scale.

To empirically validate our algorithm's efficacy, we conducted experiments using the Stanford bunny dataset[1], a standard benchmark in the field of 3D geometry processing. Due to the lack of comparable 7DOF scaled point-to-plane method, the proposed method was evaluated against the 6DOF point-to-plane algorithm. To reduce the complexity associated with non-global optimal algorithms, both Ground Truth and random values are employed as initial values for the GICP and FAST-GICP(also called VGICP). Additionally, the available plane normal information from GT is provided, thereby eliminating the need for GICP and FAST-GICP to engage in the process of nearest neighbor search and normal computation. We provide GT correspondence for the algorithm, as well as point pairings within the same plane and GT normal vectors for FAST-GICP. This approach is taken because our focus is on whether these algorithms can handle cases where the scale is not equal to 1 and whether they can achieve the correct solutions. Table II clearly demonstrates the superiority of the proposed algorithm for the case of noise level 0.01 and scale factor 0.5. Similar to $\mathcal{L}_{s1}$, algorithms for 6DOF are unable to handle data involving scale even with minimal noise. Algorithms that do not guarantee global optimality are notably susceptible to convergence to local optima. As demonstrated in [8], [28], even initializing with GT does not prevent the possibility of converging to incorrect local optima, thereby compromising the stability of the algorithm. A more fundamental issue is that methods other than ours do not support scenarios where the scale is not equal to 1. Readers are encouraged to refer to the supplement and view the illustrations for a more intuitive understanding of the data presented in the table.

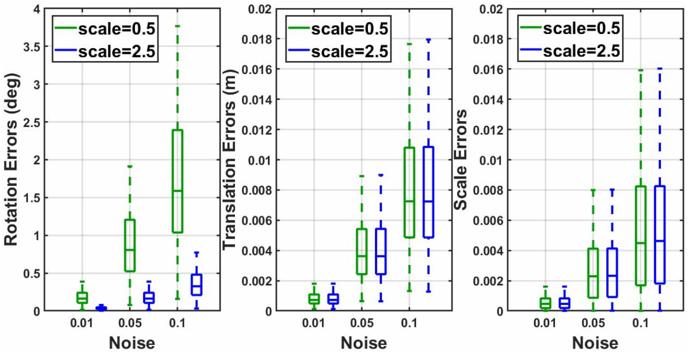


Fig. 8. $\mathcal{L}_{s_2}$ Experimental : Precision of point-to-plane matching using $\mathcal{L}_{s_2}$ under various noise levels.

We also tested different algorithms' performance at various scales with noise level 0.05, as shown in Fig. 7. Algorithms unable to solve for scale fail to yield valid results when scale is not equal to 1. In contrast, the error of our sQPEP algorithm nearly aligns with the x-axis. The translational errors of GICP, Fast-GICP, and QPEP can be attributed to their convergence to a local optimum near the ground truth. As the scale increases, their translational errors also grow linearly. For a more detailed discussion, readers are referred to the supplementary material. To better observe the error distribution of our algorithm, we evaluated its performance under various noise conditions, as shown in Fig. 8.

D. Experiments on Industrial Hand-Eye Calibration

We employ the UR10 industrial robot for experimental validation. The apparatus requiring calibration of the camera extrinsic from the camera to the end-effector is depicted in Fig. 9. We conducted four sets of hand-eye calibration tasks using this device.

[1] graphics.stanford.edu/~mdfisher/Data/Meshes/bunny.obj

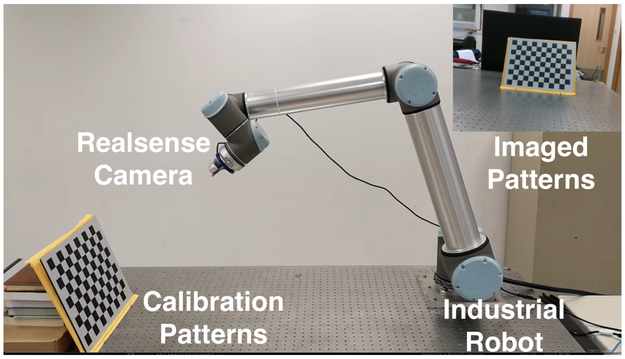


Fig. 9. Industrial Hand-Hye Calibration: Experimental setup and related devices.

TABLE III
RESULT OF INDUSTRIAL SCALE HAND-EYE CALIBRATION

		Case 1	Case 2	Case 3	Case 4
ours	ϵ_r(deg)	0.6017	0.8101	**0.0407**	**0.1086**
	ϵ_t(m)	**0.0394**	0.0098	0.0137	0.0307
Andreff [10]	ϵ_r(deg)	**0.2077**	0.1690	0.1447	0.1275
	ϵ_t(m)	0.0930	**0.0069**	**0.0090**	**0.0190**
Liang [24]	ϵ_r(deg)	1.1319	0.8916	0.7079	0.4275
	ϵ_t(m)	0.0636	0.0098	0.0138	0.0205
Wu-4D [25]	ϵ_r(deg)	1.1724	0.9058	0.7161	0.4349
	ϵ_t(m)	0.0665	0.0683	0.0694	0.0693
Sarabandi [26]	ϵ_r(deg)	5.2984	1.5769	7.1550	3.0343
	ϵ_t(m)	0.0501	0.0665	0.0575	0.0901
QPEP [9]	ϵ_r(deg)	0.2829	**0.1438**	0.0832	0.1241
	ϵ_t(m)	0.0677	0.0092	0.0093	0.0179

Our SfM process involved a systematic pipeline. We begin with camera calibration [29] to determine the intrinsic parameters. Utilizing off-the-shelf COLMAP [30], [31] software, we perform SfM to estimate relative camera poses and reconstruct the 3D scene. We use the hand-eye calibration result from perspective-n-points (PnP) as baseline. The reason is that using the PnP absolute camera poses, we verify the baseline ground truth has the reprojection error of less than 0.5 pixels, which is satisfactory to be a reference. The qualitative result is shown in Fig. 10. As the chessboard pattern used for calibration has known square sizes, the obtained results are scale-consistent. This allows for a fair comparison with 6DOF algorithms, as both approaches utilize the same calibration target. The final calibration results are summarized in Table III. As our algorithm computes $\boldsymbol{t_s}$ based on rotation, there is a slight loss in translation accuracy. Nevertheless, our method still achieves satisfactory precision when applied to real-world industrial scenarios.

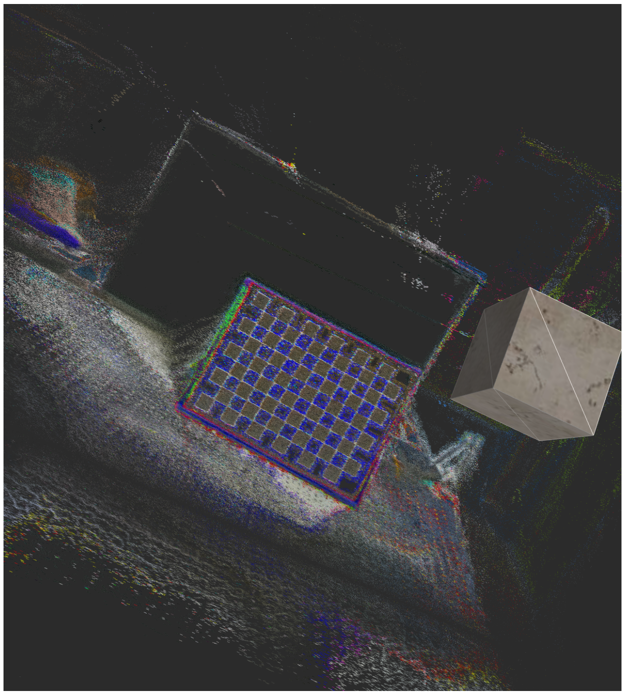


Fig. 10. Industrial Hand-Hye Calibration : The SfM result by COLMAP method [30]. The gray box denotes the initial camera pose as standard reference.

E. Experiments on EuRoc dataset

In practical industrial scenarios, monocular cameras lacking scale data might require extrinsic calibration with additional sensors installed on various platforms, including vehicles and unmanned aerial vehicles (UAVs). We employ the EuRoC dataset [32] as a benchmark to validate our approach in scenarios that closely mimic real-world conditions. Our experimental setup comprises a stereo camera system. The left camera executes vins-mono [13], capitalizing on IMU data. to maintain scale information, whereas the right camera processes imagery by employing ORB-SLAM3 [33] in a monocular configuration. The trajectories generated by both cameras are used to perform hand-eye calibration to determine the extrinsic parameters between them. This calibrated relationship is then benchmarked against the ground truth for accuracy validation. To ensure a stable evaluation, especially given the challenges associated with monocular odometry, our tests were confined to the MH01-easy, MH02-easy, and V101-easy sequences from the EuRoC dataset. Acknowledging that scale discrepancies are inherently integrated into the systems' translational and rotational errors, our assessment focuses exclusively on these two error metrics. The results, as delineated in Fig. 11, delineate the calibration discrepancies in terms of rotation (deg) and translation (m), providing a comprehensive view of the calibration precision achieved in our experimental framework.

It is crucial to highlight that the method proposed by [10] exhibits the same behavior as shown in Fig. 4. In VIO systems with high noise levels, this method fails to operate robustly, resulting in translation errors exceeding 70 meters and rotation errors surpassing 30 degrees, rendering it completely unusable in such scenarios. This underscores the significance of our globally optimal solution, which demonstrates a remarkably strong robustness against noise.

F. Runtime and Memory Evaluation

The core of our algorithm lies in the efficient computation of the solver. This subsection details the memory footprint and

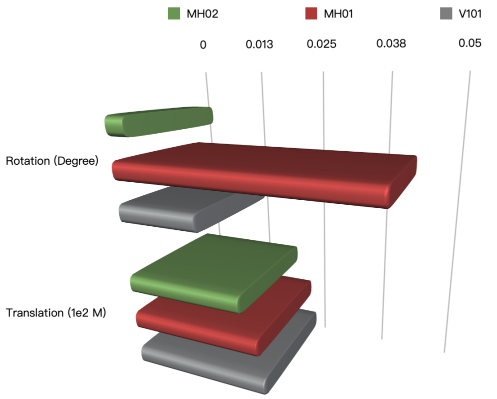


Fig. 11. EuRoC Dataset: Calibration Results for Camera Extrinsic Parameters Using the EuRoC Dataset with sequences `MH02`, `MH01`, and `V101`. Only our method yields valid numerical results; other approaches fail due to the near degeneracy of their data, rendering them ineffective.

TABLE IV
RESOURCE UTILIZATION

	Runtime(ms)	Peak Memory(MB)
Our $\mathcal{L}_{s_1}$	431	10.09
QPEP-HE [9]	1716	8.15
Andreff [10]	1654	1.25
Liang [24]	4505	1.43
Wu-4D [25]	46	8.55
Sarabandi [26]	20	4.05
Our $\mathcal{L}_{s_2}$	61	26.38
QPEP-P2P [9]	26	22.03
GICP [8]	46	0.27
Fast-GICP [28]	51	0.87

execution time during the operation of our solver, with results presented in Table IV. We use case 1 data from Section III-D for hand-eye calibration tests and data from Section III-C for point-to-plane tests. In hand-eye calibration problems, our method achieves faster speeds compared to globally optimal algorithms primarily because our elimination template requires fewer input elements and does not exhibit exponential runtime increase with the problem size. The methods in [25] and [26] are faster as they do not require computational resources to explore additional solutions for optimality. Regarding the point-to-plane scenario, the inclusion of scale in our elimination template results in a larger size compared to QPEP-P2P [9], leading to a decrease in performance. Considering our algorithm's additional capability to estimate a scale factor, it holds a distinct advantage. Memory usage is acceptable for all algorithms on modern devices.

IV. CONCLUSION

In this paper, we rigorously formalize the mathematical model for the Quadratic Pose Estimation Problem (QPEP) and present, for the first time, the framework of Scaled Quadratic Pose Estimation Problems (sQPEP), complete with precise definitions. Considering the scaling effects on rotation and translation, we delineate sQPEP into two distinct classes, providing global optimal solution for each. Experimental validation affirms our approach's robust performance in degenerate, ill-conditioned, and noisy scenarios, highlighting the inherent benefits of global solvers. Remarkably, our method inherently attains greater solution precision by improving numerical accuracy, in contrast to iterative optimization-based methods that depend on parameter settings for precision. Nonetheless, our approach necessitates manual reduction to either $\mathcal{L}s_1$ or $\mathcal{L}s_2$, and manual computation of the matrices $\boldsymbol{W}$, $\tilde{\boldsymbol{W}}$, or $\boldsymbol{Q}$, which limits its scalability. Future research should aim to reduce the solver's elimination template size, thereby decreasing the computational scale, and to broaden our method's applicability across various fields.

APPENDIX

A. Derivation of $\mathcal{L}(\boldsymbol{q},\boldsymbol{t})$

According to [9], the Quadratic Pose Estimation Problem addresses the estimation of 6DOF poses under the following constraints:

$$\begin{cases} f(\boldsymbol{q}^4,\lambda)=0\\ \boldsymbol{q}^\top\boldsymbol{q}=1\\ \boldsymbol{t}=\mathcal{T}(q^2)\end{cases} \tag{9}$$

where $\boldsymbol{t}$ must be expressible as a linear combination of $\boldsymbol{q}^2$. This necessitates that the partial derivatives of $\boldsymbol{t}$ only contain terms up to $\boldsymbol{t}^2$ to avoid introducing terms of $\boldsymbol{q}^5$. Similarly, $\boldsymbol{t}$ can only be directly multiplied by $\boldsymbol{q}^2$. After addressing the constraints on $\boldsymbol{q}$, we consider that the presence of $\boldsymbol{q}^4$ does not alter the definition provided in (9). Since $\boldsymbol{q}$ represents a rotation matrix, there are no odd powers of $\boldsymbol{q}$. Combining these terms, the variables of our function include $\boldsymbol{q}^4$, $\boldsymbol{q}^2\boldsymbol{t}$, $\boldsymbol{t}$, and $\boldsymbol{t}^2$, potentially with varying coefficients and additional constants. Thus, the expression for $\mathcal{L}(\boldsymbol{q},\boldsymbol{t})$ is given by:

$$\mathcal{L}(\boldsymbol{q},\boldsymbol{t})=\boldsymbol{V}_{35}^\top\boldsymbol{q}^4+\boldsymbol{V}_{40}^\top\mathrm{vec}(\boldsymbol{q}^2\otimes\tilde{\boldsymbol{t}})+\boldsymbol{V}_6^\top\boldsymbol{t}^2+\boldsymbol{V}_3^\top\boldsymbol{t}+C$$

B. Derivation of $\mathcal{L}_{s1}$

Regarding the addition of a scale factor to the pose's translation vector, if we substitute $\boldsymbol{t}$ with $\boldsymbol{t}^*=\boldsymbol{t}\cdot s$, the problem formulation remains unchanged, indicating that scale is not an independent variable to be solved in this scenario.

For scenarios involving sensors with unknown scale (e.g., monocular cameras), the world coordinates lack scale and are represented in a homogeneous 4×1 vector format, where the fourth dimension represents the scale factor s. Homogeneous coordinates maintain their linear characteristics after transformations by any 4×4 matrix, hence the scale information retains similar properties to other translation components: $s=\mathcal{T}_s(\boldsymbol{q}^2)$. Incorporating scale into (A) while maintaining the highest power not exceeding four, we ensure the inheritance of the QPEP properties, leading to:

$$\begin{aligned}\mathcal{L}_{s1}(\boldsymbol{q},\boldsymbol{t},s)=&\boldsymbol{V}_{35}^\top\boldsymbol{q}^4+\boldsymbol{V}_{40}^\top\mathrm{vec}(\boldsymbol{q}^2\otimes\tilde{\boldsymbol{t}})+s\cdot\boldsymbol{V}_{10}^\top\boldsymbol{q}^2+\\&\boldsymbol{V}_6^\top\boldsymbol{t}^2+s\cdot\boldsymbol{V}_4^\top\boldsymbol{t_s}+\boldsymbol{V}_4^\top\boldsymbol{t_s}+C\end{aligned}$$

C. Derivation of $\mathcal{L}_{s2}$

If the scale factor is incorporated into the rotation matrix, transforming it into a scaled rotation matrix, this change has substantial physical implications. As previously discussed, since $\boldsymbol{q}$ is derived from a rotation matrix, incorporating a scale factor s affects every occurrence of $\boldsymbol{q}^2$. The derivation, based on (9), involves multiplying each term involving $\boldsymbol{q}^2$ by s, resulting in:

$$\mathcal{L}_{s2}(\boldsymbol{q},\boldsymbol{t},s)=s^2\cdot\boldsymbol{V}_{35}^\top\boldsymbol{q}^4+s\cdot\boldsymbol{V}_{40}^\top\mathrm{vec}(\boldsymbol{q}^2\otimes\tilde{\boldsymbol{t}})+\boldsymbol{V}_6^\top\boldsymbol{t}^2+\boldsymbol{V}_3^\top\boldsymbol{t}+C.$$

REFERENCES

[1] H. Liu, T. Liu, Z. Zhang, A. K. Sangaiah, B. Yang, and Y. Li, "Arhpe: Asymmetric relation-aware representation learning for head pose estimation in industrial human–computer interaction," *IEEE Trans. Ind. Inform.*, vol. 18, no. 10, pp. 7107–7117, 2022.

[2] S. Cheng, C. Sun, S. Zhang, and D. Zhang, "Sg-slam: A real-time rgb-d visual slam toward dynamic scenes with semantic and geometric information," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2022.

[3] B. Xue, X. Yan, J. Wu, J. Cheng, J. Jiao, H. Jiang, R. Fan, M. Liu, and C. Zhang, "Visual-marker-based localization for flat-variation scene," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–16, 2024.

[4] J. Yuan, S. Zhu, K. Tang, and Q. Sun, "Orb-tedm: An rgb-d slam approach fusing orb triangulation estimates and depth measurements," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.

[5] L. Zhou, D. Koppel, and M. Kaess, "A complete, accurate and efficient solution for the perspective-n-line problem," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 699–706, 2020.

[6] J. Wu, M. Liu, C. Zhang, and Z. Zhou, "Correspondence matching and time delay estimation for hand-eye calibration," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8304–8313, 2020.

[7] C. Lv, W. Lin, and B. Zhao, "Kss-icp: point cloud registration based on kendall shape space," *IEEE Transactions on Image Processing*, vol. 32, pp. 1681–1693, 2023.

[8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[9] J. Wu, Y. Zheng, Z. Gao, Y. Jiang, X. Hu, Y. Zhu, J. Jiao, and M. Liu, "Quadratic pose estimation problems: Globally optimal solutions, solvability/observability analysis, and uncertainty description," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3314–3335, 2022.

[10] N. Andreff, R. Horaud, and B. Espiau, "Robot hand-eye calibration using structure-from-motion," *Int. J. Rob. Res.*, vol. 20, no. 3, pp. 228–248, 2001.

[11] S. Song, M. Chandraker, and C. C. Guest, "High accuracy monocular sfm and scale correction for autonomous driving," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 730–743, 2016.

[12] S. Roos-Hoefgeest, I. A. Garcia, and R. C. Gonzalez, "Mobile robot localization in industrial environments using a ring of cameras and aruco markers," in *IEEE IECON 2021*, 2021, pp. 1–6.

[13] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[14] Z. Wang, H. Dai, Y. Zeng, and T. C. Lueth, "A robust 6-d pose tracking approach by fusing a multi-camera tracking device and an ahrs module," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2021.

[15] J. Hu, W. Lei, D. Zhuo, F. Zhu, W. Liu, and X. Zhang, "Enhance pose and extrinsic accuracy with online spatial and temporal compensation in monocular camera-aided gnss/sins integration," *IEEE Trans. Instrum. Meas.*, 2024.

[16] T. Du, S. Shi, Y. Zeng, J. Yang, and L. Guo, "An integrated ins/lidar odometry/polarized camera pose estimation via factor graph optimization for sparse environment," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[17] C. L. Glennie, A. Kusari, and A. Facchin, "Calibration and stability analysis of the vlp-16 laser scanner," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XL-3/W4, pp. 55–60, 2016.

[18] S. Ying, J. Peng, S. Du, and H. Qiao, "A scale stretch method based on icp for 3d data registration," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 3, pp. 559–565, 2009.

[19] J. Heller, M. Havlena, and T. Pajdla, "Globally optimal hand-eye calibration using branch-and-bound," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 1027–1033, 2015.

[20] J. Wu, M. Liu, Y. Zhu, Z. Zou, M.-Z. Dai, C. Zhang, Y. Jiang, and C. Li, "Globally optimal symbolic hand-eye calibration," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1369–1379, 2020.

[21] V. Larsson, K. Astrom, and M. Oskarsson, "Efficient solvers for minimal problems by syzygy-based reduction," in *IEEE ICCV*, 2017, pp. 820–829.

[22] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *ECCV*. Springer, 2008, pp. 302–315.

[23] C. Stachniss, J. Leonard, and S. Thrun, *Springer Handbook of Robotics, 2nd edition*, 2016.

[24] R.-h. Liang and J.-f. Mao, "Hand-eye calibration with a new linear decomposition algorithm," *J. Zhejiang Univ.-Sci. A*, vol. 9, no. 10, pp. 1363–1368, 2008.

[25] J. Wu, Y. Sun, M. Wang, and M. Liu, "Hand-eye calibration: 4-d procrustes analysis approach," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 2966–2981, 2019.

[26] S. Sarabandi, J. M. Porta, and F. Thomas, "Hand-eye calibration made easy through a closed-form two-stage method," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3679–3686, 2022.

[27] J. Jiang, X. Luo, Q. Luo, L. Qiao, and M. Li, "An overview of hand-eye calibration," *The International Journal of Advanced Manufacturing Technology*, vol. 119, no. 1, pp. 77–97, 2022.

[28] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized gicp for fast and accurate 3d point cloud registration," in *2021 IEEE ICRA*. IEEE, 2021, pp. 11 054–11 059.

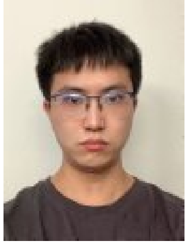
[29] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[30] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE CVPR*, 2016, pp. 4104–4113.

[31] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV*, 2016, pp. 501–518.

[32] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *Int. J. Rob. Res.*

[33] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE trans. robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.



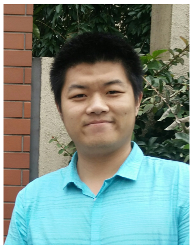
Bohuan Xue received the B.Eng. degree in computer science and technology from College of Mobile Telecommunications, Chongqing University of Posts and and Telecom, Chongqing, China, in 2018, and the Ph.D. degree from the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Hong Kong, China, in 2024. He is currently working as research assistant in School of Data Science and Engineering, and Xingzhi College, South China Normal University.

Yilong Zhu was born in 1994. He received the B.Sc. degree in 2017 from the Harbin Institute of Technology, and the M.S. degree in electrical engineering in 2018 from the Hong Kong University of Science and Technology, Hong Kong, where he is currently working toward the Ph.D. degree, supervised by Prof. Shaojie Shen. His current research interests include LiDAR and sensor fusion for robotics.

Tianyu Liu obtained his B.S. degree from Harbin Institute of Technology in Harbin, China, and his M.S. degree from Northwestern University in Shenyang, Liaoning, China. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, HKUST, Hong Kong, supervised by Prof. Ping Tan. His research interests include neural implicit representation and autonomous system perception.

Jin Wu (Member, IEEE) received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China. He is currently a PhD student with HKUST. In 2013, He was a visiting student in Groep T, KU Leuven. From 2019 to 2020, He was with Tencent Robotics X. He has published over 140 papers in representative journals and conferences. He was named in Stanford University List of Top 2% Scientists Worldwide in 2020, 2021, 2022, according to contributions in Aerospace Engineering and Robotics.

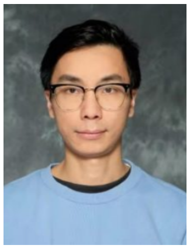
Jianhao Jiao (Member, IEEE) received the B.Eng. degree in instrument science from Zhejiang University, Hangzhou, China, in 2017, and the Ph.D. from the Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, Hong Kong, China, in 2021. He is currently a senior research fellow With the Department of Computer Science, University College London.



Yi Jiang (Member, IEEE) was born in Ezhou, Hubei, China. He received the B.E., M.S., and Ph.D. degrees from Northeastern University, Shenyang, Liaoning, China. He is currently a Postdoctoral Fellow with the City University of Hong Kong, Hong Kong. Dr. Jiang is the recipient of the Excellent Doctoral Dissertation Award from Chinese Association of Automation in 2021. He is an Associate Editor of Advanced Control for Applications: Engineering and Industrial Systems (Wiley).

Chengxi Zhang (Member, IEEE) was born in Feb, 1990, Shandong, China. He received his B.S. and M.S. degrees in microelectronics and solid-state electronics at Harbin Institute of Technology, China, in 2012 and 2015. He received his Ph.D. degree in control science and engineering at Shanghai Jiao Tong University, China, in 2019. He is an Associate Professor with School of Internet of Things, Jiangnan University, Wuxi, China. His research interests are embedded system software and hardware design, information fusion and control theory.

Xinyu Jiang received the B.E. degree in mechanical engineering and automation and software engineering from Dalian Jiaotong University, China, in 2016, and the master's degree in computer science from the University of Macau, Macau, China, in 2021. He has been working toward the Ph.D. degree in Computer Science at the University of Macau, Macau, China, since 2022.



Zhijian He received the B.E. and M.E. degrees in the Department of Electronic Science and Technology Engineering, Jilin University, China, in 2009 and 2013, respectively, and the Ph.D. degree in Department of Computing, the Hong Kong Polytechnic University, in 2019. As serving as a Research Associate in The Hong Kong University of Science and Technology, meanwhile, he is an Assistant Professor in College of Big Data and Internet, Shenzhen Technology University (SZTU).