
A Direct Least-Squares Method for the PnP Problem in 2D Forward-Looking Sonar

Journal:	<i>IEEE Sensors Journal</i>
Manuscript ID	Sensors-81169-2024
Manuscript Type:	Regular Paper (S1)
Date Submitted by the Author:	31-Oct-2024
Complete List of Authors:	Su, Jiayi; Zhejiang University, Ocean College Zhu, Haodi; Zhejiang University, Ocean College Qian, Jingyu ; Zhejiang University, Ocean College Yuan, Yufan Qu, Fengzhong; Zhejiang University, Ocean Engineering Wei, Yan; Zhejiang University, Ocean College
EDICS:	DATP

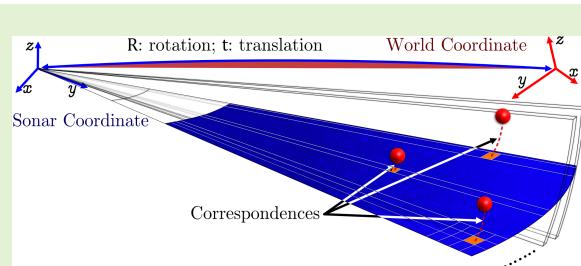
SCHOLARONE™
Manuscripts

A Direct Least-Squares Method for the PnP Problem in 2D Forward-Looking Sonar

Jiayi Su , Haodi Zhu , Jingyu Qian , Yufan Yuan ,
Fengzhong Qu  Senior Member, IEEE, and Yan Wei  Member, IEEE

Abstract— Solving the perspective-n-point (PnP) problem involves estimating the sensor pose given 3D points in world coordinates and their corresponding 2D measurements. This problem is a common pose estimation problem that has been well-investigated in the field of conventional optical cameras, but it is still underdeveloped when using 2D forward-looking sonar (2D FLS) as the imaging sensor. We propose a direct least-squares (DLS) approach, named DLSS, to address the PnP problem in 2D FLS. This method offers several advantages: 1) It achieves higher precision than the previous state-of-the-art (SOTA) in rotation estimation, while offering comparable performance in translation estimation; 2) It provides a unified solution framework for both general and coplanar scenarios; 3) The solution for t_z is in closed form, which accelerates the solving process, making the solver 10 times faster than the previous SOTA method. The simulation experiments are conducted to validate the effectiveness of the proposed method.

Index Terms— Pose estimation, perspective-n-point (PnP) problem, forward-looking sonar (FLS), Gröbner basis.



I. INTRODUCTION

THE pose estimation problem is important for autonomous underwater vehicles (AUVs) during underwater surveying tasks. Although optical cameras are widely adopted by AUVs to provide basic visual capabilities due to their well-studied imaging principles and low expense [1], their visibility is easily affected by harsh underwater conditions, such as the selective absorption of light by water, marine snow, and the need for external light sources, thereby making pose estimation less reliable [2]. The 2D forward-looking sonar (FLS) can overcome the aforementioned issues, providing longer-range and more stable environmental measurements, making it widely applied in underwater applications [3]–[5]. However, due to its fundamentally different imaging mechanism compared with optical cameras, even though it can provide photo-realistic acoustic images, directly applying algorithms designed for optical cameras to solve the pose estimation problem is infeasible. The perspective-n-point (PnP) problem for 2D FLS is precisely such a case.

PnP is the problem of estimating the pose given a set of n 3D points in world coordinates and their corresponding 2D

measurements in the image [6]. The applications are widely employed in robotics and computer vision tasks, such as simultaneous localization and mapping (SLAM) [7], structure-from-motion (SfM) [8] and augmented reality (AR) [9]. The PnP problem is extensively researched in the optical camera community and has many mature algorithms, from the earlier [10]–[12], to the recent [13]–[15]. Despite potential numerical problems or efficiency drawbacks, the solver based on the Gröbner basis method remains a standard for comparison [12], maintaining competitive performance. Therefore, in this paper, we utilize the Gröbner basis to directly solve the derived polynomial system in its least-squares form, ensuring the optimality of the results.

While being well-studied for optical cameras, the PnP problem for 2D FLS in underwater scenarios remains underdeveloped. We briefly introduce the related works. In [16], [17], iterative methods are applied to estimate the pose of the 2D FLS relative to planar grid-like targets. A good initial guess is essential, and the method is prone to getting stuck in local minima. By first acquiring a closed-form solution and then refining it through iterative optimization, [18]–[20] can obtain stable and accurate pose estimation results. But iterative methods introduce considerable computational overhead, and determining the translation along z -axis also depends on iterative refinement, increasing the overall computational expense. [21] can solve t_z analytically, but it assumes a planar point configuration and easily encounters singular situations (when solving t_z). Another class of PnP-like problems involves using a known model instead of just point information to estimate the sonar's poses [22], [23]. Although these methods are more

Manuscript received placeholder.

This work was supported by the Key Research and Development Project of Hainan Province under Grant ZDYF2023GXJS010.

Jiayi Su, Haodi Zhu, Jingyu Qian, Yufan Yuan, Fengzhong Qu, and Yan Wei are with the Ocean College, Zhejiang University, Zhoushan 316021, China. (e-mail: sujyzju, haodizhu, qianjingyu, y_yuf, jinmufz, redwine447@zju.edu.cn)

Fengzhong Qu is also with the Hainan Institute of Zhejiang University, Sanya 572025, China.

Digital Object Identifier placeholder.

robust, they require a thorough prior understanding of the scene.

In this paper, we propose DLSS (Direct Least-Squares for Sonar). The method is inspired by [11], where the original formulation is non-approximated, and we found it to be highly unstable. After applying the weak-perspective approximation, both performance and stability are significantly improved. Additionally, a closed-form solution for t_z is proposed, which substantially improves the solving speed while maintaining accuracy comparable to optimization-based methods. Our contributions are summarized as follows:

- 1) We propose the DLSS, a non-iterative solver for the PnP problem in 2D FLS. It outperforms previous state-of-the-art (SOTA) methods in rotation estimation, while achieving comparable results in translation estimation.
- 2) The DLSS provides a unified solution framework for both general and coplanar cases, eliminating the need for users to design additional algorithms specifically for the coplanar case.
- 3) Extensive simulation experiments, accompanied by thorough analysis and discussions, are conducted to validate the efficiency of the DLSS.

The balance of the paper is organized as follows: Some preliminaries are introduced in Sec. II. The proposed methods are detailed in Sec. III. Sec. IV presents simulation experiments, along with discussions. Finally, conclusions are provided in Sec. V.

II. PRELIMINARIES

A. The Projection Model of 2D FLS

¹ A 3D point \mathbf{P}^s in sonar coordinates can be described using (r, θ, ϕ) in spherical coordinates:

$$\mathbf{P}^s = \begin{bmatrix} X^s \\ Y^s \\ Z^s \end{bmatrix} = \begin{bmatrix} r \cos \phi \sin \theta \\ r \cos \phi \cos \theta \\ r \sin \phi \end{bmatrix}, \quad (1)$$

where r is the measured distance between the \mathbf{P}^s and the center of the sonar's transmitting array, θ is the bearing angle, ϕ is the elevation angle. During measurements, ϕ is lost, causing *elevation ambiguity*. Thus, the corresponding projected 2D point \mathbf{p}^s can be obtained through

$$\mathbf{p}^s = \begin{bmatrix} x^s \\ y^s \end{bmatrix} = \begin{bmatrix} r \sin \theta \\ r \cos \theta \end{bmatrix}. \quad (2)$$

For each point i , combining Eqs. 1 and 2, we get

$$\begin{bmatrix} x_i^s \\ y_i^s \end{bmatrix} = \begin{bmatrix} \cos^{-1} \phi_i & 0 & 0 \\ 0 & \cos^{-1} \phi_i & 0 \end{bmatrix} \begin{bmatrix} X_i^s \\ Y_i^s \\ Z_i^s \end{bmatrix}. \quad (3)$$

For sonars from different manufacturers, the typical range of ϕ values is approximately 12° to 20° , which results in $\cos \phi \in [0.93969, 1]$. For some problems in the fields of computer vision and robotics, researchers have linearized the term $\cos \phi$ to simplify the analysis process by assuming $\cos \phi = \alpha$. If $\alpha = 1$, the projection is approximated as an

¹For the PnP problem in 2D FLS, we focus on the geometric projection characteristics. For details on its working mechanism, please refer to [24].

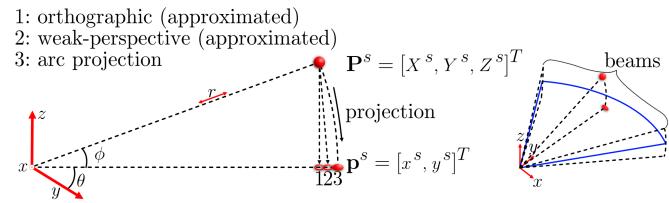


Fig. 1: The projection model of 2D FLS. The left part is a side view of one of the dashed sectors in the right part. The arc projection represents the true working mechanism of the sonar.

orthographic model [4], [25]; if $\alpha \neq 1$, it is approximated as a weak-perspective model [18]. The projection characteristics are concluded in Fig. 1.

B. Problem Formulation

The relationship between \mathbf{P}^w in world coordinates and the corresponding measurements \mathbf{p}^s can be described as follows:

$$\underbrace{\begin{bmatrix} x_i^s \\ y_i^s \\ 1 \end{bmatrix}}_{\mathbf{p}_i^s} = \underbrace{\begin{bmatrix} \cos^{-1} \phi_i & 0 & 0 & 0 \\ 0 & \cos^{-1} \phi_i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{proj}} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{T}_w^s} \underbrace{\begin{bmatrix} X_i^w \\ Y_i^w \\ Z_i^w \\ 1 \end{bmatrix}}_{\mathbf{P}_i^w}, \quad (4)$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the orientation and position to be determined, \mathbf{T}_w^s is composed of \mathbf{R} and \mathbf{t} to represent the homogeneous coordinate transformation from world to sonar frame, proj represents the projection function(or matrix). To solve for \mathbf{R} and \mathbf{t} , a common approach is to aggregate each pair $\{(\mathbf{P}_i^w, \mathbf{p}_i^s) \mid i = 1, 2, \dots, N\}$ in the relationship described by Eq. 4 into least-square form:

$$\begin{aligned} \arg \min_{\mathbf{R}, \mathbf{t}} \sum_{i=1}^N & \| \text{proj}(\mathbf{R}, \mathbf{t}, \mathbf{P}_i^w) - \mathbf{p}_i^s \|_2^2, \\ \text{s.t. } & \forall i, \quad \phi_{min} \leq \phi_i \leq \phi_{max}, \\ \phi_i &= \frac{\sqrt{(X_i^s)^2 + (Y_i^s)^2}}{\sqrt{(X_i^s)^2 + (Y_i^s)^2 + (Z_i^s)^2}}. \end{aligned} \quad (5)$$

Due to the non-convexity and nonlinearity in Eq. 5, gradient-based iterative optimization methods, such as Gauss-Newton or LevenbergMarquardt, are prone to getting trapped in local minima [26]. To avoid this issue, we reformulate Eq. 5 into a polynomial system and solve it using the Gröbner basis technique in Sec. III.

C. Gröbner Basis Solver

Many computer vision problems can be expressed as solving systems of multivariable polynomials. A well-known method for solving polynomial system is to compute the Gröbner basis (GB), which has the same solutions as the original problem but is easier to solve. Early work involved manually constructing the basis [27], which required researchers to have a deep knowledge of algebraic geometry. [28] firstly proposed an automatic generator of GB solvers, expanded the application in computer vision community. The process of generator

includes determining the Gröbner basis and monomial bases, constructing the elimination template and building the action matrix from it. The solutions are extracted from the eigen-factorization of the action matrix. Readers can refer to [29] for detailed information on solving polynomials using the Gröbner basis.

III. THE PROPOSED METHOD

A. Reformulating as a Polynomial System

To solve using a GB solver, we first formulate the PnP problem in 2D FLS as a polynomial system. From Eq. 4, we can get

$$\cos \phi_i \begin{bmatrix} x_i^s \\ y_i^s \end{bmatrix} - \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \mathbf{R}_{1:2} \begin{bmatrix} X_i^w \\ Y_i^w \\ Z_i^w \end{bmatrix}, \quad (6)$$

where $\mathbf{R}_{1:2}$ is the top two rows of \mathbf{R} , t_x and t_y are two components in \mathbf{t} . Considering all i from 1 to N , we obtain

$$\underbrace{\begin{bmatrix} x_1^s & \dots & -1 & 0 \\ y_1^s & \dots & 0 & -1 \\ \ddots & & \vdots & \\ x_N^s & -1 & 0 \\ y_N^s & 0 & -1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \cos \phi_1 \\ \vdots \\ \cos \phi_N \\ t_x \\ t_y \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{R}_{1:2} & & \\ & \ddots & \\ & & \mathbf{R}_{1:2} \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} \mathbf{P}_1^w \\ \vdots \\ \mathbf{P}_N^w \end{bmatrix}}_{\mathbf{b}}, \quad (7)$$

where $\mathbf{A} \in \mathbb{R}^{2N \times (N+2)}$ and $\mathbf{b} \in \mathbb{R}^{3N}$ are known, $\mathbf{x} \in \mathbb{R}^{N+2}$ contains the unknowns we want to eliminate, $\mathbf{W} \in \mathbb{R}^{2N \times 3N}$ is a block diagonal matrix of the $\mathbf{R}_{1:2}$. We can get $\mathbf{x} = \mathbf{A}^\dagger \mathbf{W} \mathbf{b}$, where \mathbf{A}^\dagger is the MoorePenrose inverse of \mathbf{A} . By partitioning \mathbf{A}^\dagger into $[\mathbf{U}, \mathbf{V}]^T$, we have

$$\cos \phi_i = \mathbf{u}_i^T \mathbf{W} \mathbf{b}, \quad (8)$$

$$\begin{bmatrix} t_x \\ t_y \end{bmatrix} = \mathbf{V} \mathbf{W} \mathbf{b}, \quad (9)$$

where \mathbf{u}_i^T is the i -th row of \mathbf{U} . The derived result has a similar form to [11], with the difference being 1) the left side of Eq. 8 is not the distance between the point and the camera, but rather $\cos \phi$, which has a more stringent range of values; 2) the \mathbf{W} is composed of only the first two rows of \mathbf{R} . Substituting Eqs. 8 and 9 into Eq. 6, we obtain an equation containing only $\mathbf{R}_{1:2}$ as the unknowns:

$$\mathbf{u}_i^T \mathbf{W} \mathbf{b} p_i^s - \mathbf{V} \mathbf{W} \mathbf{b} = \mathbf{R}_{1:2} \mathbf{P}_i^w. \quad (10)$$

Note that both sides of the Eq. 10 are 2×1 vectors, which means one point gives two constraints. For each constraint in Eq. 10, we can obtain $\mathbf{c}_i^T \mathbf{m} = 0$, where \mathbf{m} and \mathbf{c} represent the monomials and their corresponding coefficients, respectively. For $i = 1, \dots, N$, we obtain

$$\mathcal{L} = \mathbf{m}^T \mathbf{S} \mathbf{m}, \quad (11)$$

$$\mathbf{S} = \mathbf{C}^T \mathbf{C}, \quad (12)$$

where each row of \mathbf{C} is \mathbf{c}_i^T . The Eq. 11 is a concise representation of the cost function we aim to minimize. As in [11], we use the Cayley-Gibbs-Rodriguez (CGR) parametrization to represent \mathbf{R} , as it has the most compact representation for the rotation matrix [30]. By CGR parameterization, we have

$$\mathbf{R} = \frac{(1 - \mathbf{s}^T \mathbf{s}) \mathbf{I}_3 + 2[\mathbf{s} \times] + 2\mathbf{s}\mathbf{s}^T}{1 + \mathbf{s}^T \mathbf{s}}, \quad (13)$$

where $\mathbf{s} = [s_1, s_2, s_3]^T$ are CGR parameters, $[\mathbf{s} \times]$ denotes the operation of converting a vector into a skew-symmetric matrix. Substituting Eq. 13 back into Eq. 10 and multiplying both sides by $1 + \mathbf{s}^T \mathbf{s}$, we find that \mathcal{L} is a polynomial of degree four in three variables.

In practice, we found that such a formulation leads to unstable results. As pointed out in [20], the constraints on each $\cos \phi_i$, namely $\mathbf{u}_i^T \mathbf{W} \mathbf{b}$, are of significance to ensure a stable and valid solution. However, Eq. 11 does not contain such constraints. If inequality constraints are added, solving Eq. 11 would revert to using iterative methods for optimization, instead of being able to use a Gröbner basis-based solver to obtain a direct solution. A possible workaround is to check valid solutions with the Karush-Kuhn-Tucker (KKT) conditions [31], which often results in an empty solution set when constraints are always active under severe noise conditions. To overcome this issue, we propose to approximate $\cos \phi_i$ for each i as a constant α , using a weak-perspective model, which is similar to [18]. Notably, in [18], the points need to be coplanar, whereas the proposed method does not have this requirement.

The approximation has its geometric significance, considering all points to be located on the $\bar{\phi}$ plane, where $\alpha = \cos \bar{\phi}$. In practice, we found that this approximation can greatly improve the stability of the solver. Revisiting Eq. 10, by replacing $\mathbf{u}_i^T \mathbf{W} \mathbf{b}$ with α , we get

$$\alpha \mathbf{p}_i^s - \mathbf{V} \mathbf{W} \mathbf{b} = \mathbf{R}_{1:2} \mathbf{P}_i^w. \quad (14)$$

Following similar steps, we can obtain \mathcal{L}' and \mathcal{S}' . \mathcal{L}' is also a polynomial of degree four in three variables.

B. Solving the Polynomial System

To find the minimum of \mathcal{L}' , we seek its first-order optimality conditions:

$$\frac{\partial \mathcal{L}'}{\partial s_1} = 0, \quad \frac{\partial \mathcal{L}'}{\partial s_2} = 0, \quad \frac{\partial \mathcal{L}'}{\partial s_3} = 0, \quad \frac{\partial \mathcal{L}'}{\partial \alpha} = 0. \quad (15)$$

The Eq. 15 are polynomials of degree three in three variables. It is important to note that in order to eliminate the denominator in $\mathbf{R}_{1:2}$ in Eq. 14, the left multiplier of \mathbf{p}_i^s becomes $\alpha(1 + \mathbf{s}^T \mathbf{s})$. We found that this leads to generating a large elimination template, and extremely time consuming. To mitigate this issue, we introduce a substituted variable $\alpha' = \alpha(1 + \mathbf{s}^T \mathbf{s})$ during implementation. By avoiding the coupling between the α and \mathbf{s} , the number of monomials is reduced. The resulting elimination template size is 157×184 , providing faster solver generation and more stable solutions. The α can be easily obtained by dividing α' by the estimated $1 + \hat{\mathbf{s}}^T \hat{\mathbf{s}}$. For clarity, we will not distinguish between α and α' in the following text. The number of critical points of the \mathcal{L}'

are at most 27, bounded by Bézout theorem² [29]. Through proj , we select the one with the minimum reprojection error as the final solution.

C. Solving for \mathbf{R} and \mathbf{t}

When \mathcal{L}' is minimized, the complete \mathbf{R} can be directly derived using Eq. 13 and the estimated $\hat{\mathbf{s}}$ without the need to specifically consider orthogonality and determinant constraints. For (t_x, t_y) , the values can be derived through the application of Eq. 9. However, in practice, we found that, as done in [20], randomly selecting an observed \mathbf{P}^w as the origin and using its observed value as the estimated (t_x, t_y) is both stable and accurate. We use this method as the final approach for obtaining the DLSS (t_x, t_y) result, additionally also providing the results and analysis of Eq. 9 in Sec. IV. While for t_z , further effort is needed.

In [21], t_z is solved analytically using only one point a time with the requirement that the points lie on a planar target. Additionally, it is necessary to handle singular cases where no solution exists. While our closed-form solution uses all the available points without the coplanar requirement, making it more robust and almost free from singularities, and is fast comparing with the iterative method adopted in [18], [20].

For each i , we have the following equation:

$$r_i = \sqrt{(X_i^s)^2 + (Y_i^s)^2 + (Z_i^s)^2} = \sqrt{(x_i^s)^2 + (y_i^s)^2}. \quad (16)$$

By substituting \mathbf{R} , \mathbf{P}_i^w and \mathbf{p}_i^s into Eq. 16 and converting it into least-squares form, we obtain

$$\sum_{i=1}^N \|(\mathbf{r}_1 \mathbf{P}_i^w + t_x)^2 + (\mathbf{r}_2 \mathbf{P}_i^w + t_y)^2 + (\mathbf{r}_3 \mathbf{P}_i^w + t_z)^2 - (\mathbf{p}_i^s)^2\|, \quad (17)$$

where \mathbf{r}_k is the k -th row of \mathbf{R} . The only unknown in Eq. 17 is t_z , resulting in a univariate polynomial of degree four. We denote Eq. 17 by \mathcal{L}_z . Examining \mathcal{L}_z , the coefficient of t_z^4 is always 1, ensuring the existence of a global minimum of \mathcal{L}_z . We find all real roots where $d\mathcal{L}_z/dt_z = 0$, check if $d^2\mathcal{L}_z/dt_z^2 > 0$, then identify the one that minimizes \mathcal{L}_z as \hat{t}_z . The root-finding step is achieved using Matlab command `roots` with minimal overhead.

IV. EXPERIMENTS AND RESULTS

A. The Settings of Simulations

We follow the same simulation setup as in [20] for convenient comparison. 3D points were randomly generated within a $[-0.6, 0.6] \times [1.6, 2.8] \times [-0.3, 0.3]$ box in the sonar coordinates. We randomly select the coordinates of a point as \mathbf{t}^{gt} . For \mathbf{R}^{gt} , we randomly generate a quaternion with a real part no less than $\cos(a/2)$, restricting the rotation angle within $a \in [-90^\circ, 90^\circ]$, then convert the quaternion to a rotation matrix³.

²In practice, the prerequisite of Bézout theorem is usually met [11].

³It is well-known that the CGR parameterization is singular when rotating 180° around any axis, as the corresponding quaternion always has a real part of 1. Despite this limitation, the distribution of the relative positions between the points and the sonar is not constrained. Additionally, a range of $[-90^\circ, 90^\circ]$ is sufficiently broad in practice. In most cases, the rotation angle can be determined within a smaller range using prior knowledge of the environment.

Each experiment runs for 300 trials for analysis. For planar cases, we force the plane to pass through a point with $Z^s = 0$ and X^s and Y^s within the bounds $[-0.15, 0.15] \times [2.05, 2.35]$ to avoid singular configuration. The angle between the plane and the xy -plane is constrained to be between 5° and 70° , as this represents a reasonable observation angle in practice. Denoting the normal of the plane by $\mathbf{n} = [n_x, n_y, n_z]^T$, we impose the condition $(n_y \cdot n_z < 0 \wedge n_x \cdot n_z > 0)$ to avoid the dual-pose problem⁴ [21]. We also perform singular value decomposition (SVD) on the generated points, and impose constraints on singular values to avoid degenerate cases caused by collinearity. The simulation was run on a computer with an Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, 16.0 GB of RAM, and Windows 11 22H2, using Matlab R2021b. For rotation error metrics, we use $\max_{k=1}^3 \arccos(\mathbf{r}_k^{gt} \hat{\mathbf{r}}_k^T)/\pi \times 180$. For translation, we separate the evaluation of $\mathbf{t}_{xy} = [\hat{t}_x, \hat{t}_y]^T$ and \hat{t}_z as same in [20]. We use $\|\hat{\mathbf{t}}_{xy} - \mathbf{t}_{xy}^{gt}\|/\|\mathbf{t}_{xy}^{gt}\|$ and $\|\hat{t}_z - t_z^{gt}\|/\|\mathbf{t}_{xy}^{gt}\|$.

For comparison, we replicated the **Approximated** method proposed in [20], named Wang's, as it provides the best initial value for PnP problems in most cases. In [20], t_z is obtained through optimization. We replace it with our proposed method for solving t_z , called Wang's (+Our t_z). The proposed method is named DLSS. If the solution for t_z is replaced with the method from [20], it is referred to as DLSS (+ Optimized t_z). Lastly, as mentioned in Sec. III-C, we provide the results obtained from calculating \mathbf{t}_{xy} using Eq. 9, called DLSS (+Original \mathbf{t}_{xy}).

B. General Case

Firstly, we analyze the performance of 10 points under different noise levels. Gaussian noise with different standard deviations was added to the observed pixels. The results are shown in the first row of Fig. 2.

The accuracy of all methods decreases as noise increases. For rotation estimation, DLSS outperforms Wang's. We attribute this to the CGR parameterization, which ensures that the resulting solution naturally lies on $SO(3)$. In contrast, in [20], an SVD projection is required. For \mathbf{t}_{xy} estimation, DLSS (+Original \mathbf{t}_{xy}) gives better estimates under low noise levels. However, the performance quickly deteriorates as noise increases. We suppose this is because Eq. 9 expresses \mathbf{t}_{xy} in terms of VWB, making the formulation more inclined to select an appropriate estimation of $\mathbf{R}_{1:2}$ that minimizes the cost function, rather than focusing directly on \mathbf{t}_{xy} itself. For t_z estimation, optimization-based methods provide the best performance, with DLSS (+ Optimized t_z) performing better under low noise levels. The proposed closed-form method for solving t_z shows similar performance to the optimization-based method under low noise but slightly decreases as noise increases, indicating its effectiveness. On the other hand, being analytical, it offers significantly faster

⁴The constraint arises from the assumption that, in most cases, we can always determine which side of the plane is being observed. The authors also want to clarify that this constraint stems from the inherent ambiguity of the 2D FLS PnP problem, and no additional treatment is required within the methods themselves.

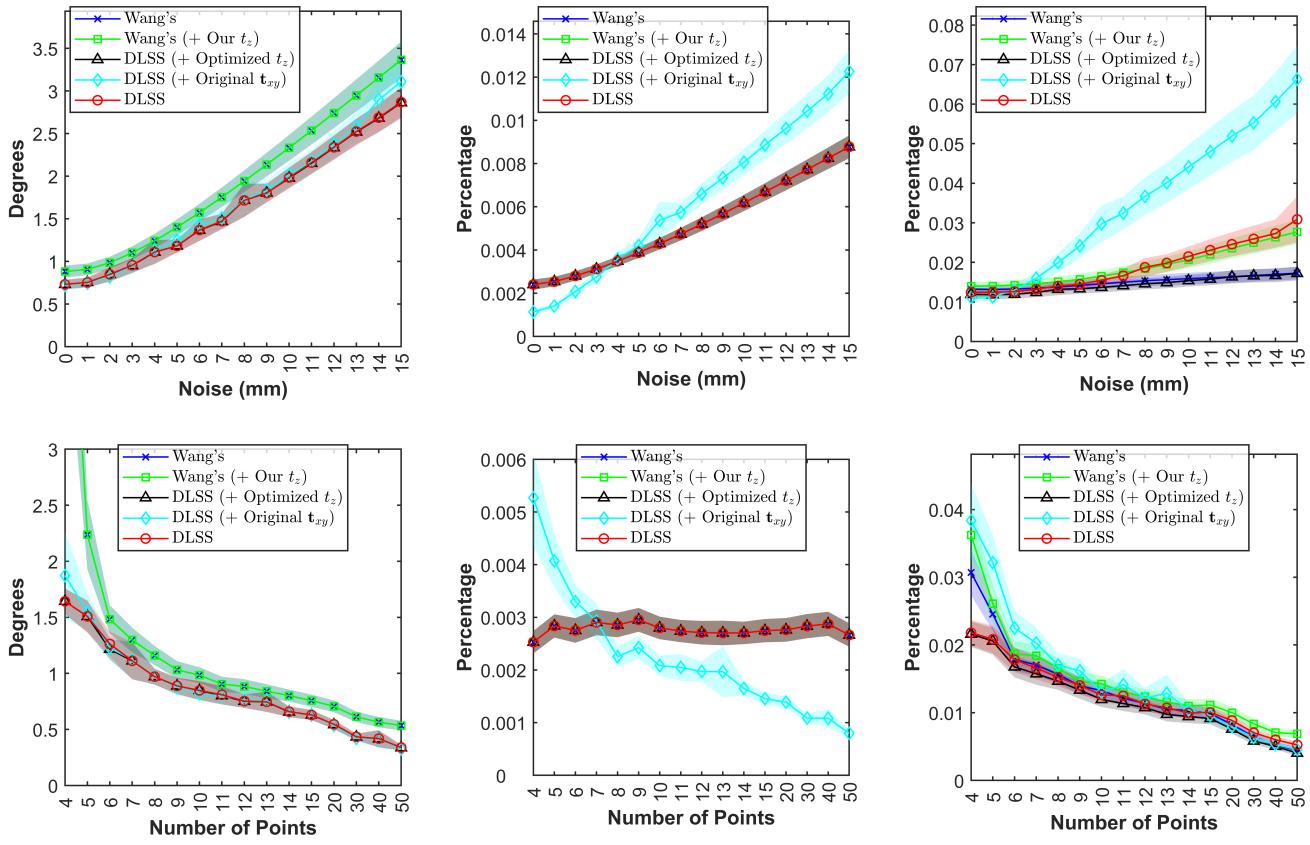


Fig. 2: The results of the general case. The first row represents the results with 10 points under different noise levels. The second row represents the results with 2mm noise under different numbers of points. The first column represents the angular error, the second column represents the t_{xy} error, and the third column represents the t_z error. The semi-transparent area indicates the 95% confidence interval. The same applies to Fig. 3.

solving speed, which provides substantial application advantages. DLSS(+Original t_{xy}) experiences rapid accuracy degradation as noise increases, since the translation estimation is sequential and is affected by the inaccurate estimation of t_{xy} .

When the noise level is set to 2mm and the number of points is varied, all methods show improved performance as the number of points increases, with performance differences between methods similar to those previously mentioned. Notably, for t_{xy} estimation, DLSS(+Original t_{xy}) shows a continuous improvement in accuracy as the number of points increases, similar to its superior performance under low noise levels.

Overall, DLSS provides better performance for rotation estimation, and comparable precisions for translation estimation. Especially for t_z , it slightly sacrifices accuracy under high noise levels but achieves a significant improvement in speed, which will be analyzed in Sec. IV-E.

C. Coplanar Case

In [20], the sign of t_z is considered to be given. We did not make any assumptions about t_z , but instead constrained the normal vector of the plane. The effects of these two approaches are similar.

For the coplanar case, as shown in Fig. 3, regardless of varying the number of points or the noise level, the trend and relative performance of different methods are similar to the general case. Overall, the performance of the methods decreases in the coplanar case because the constraints provided by coplanar points are weaker compared with the general case. An exception is the t_{xy} estimation method proposed in [20], as the process is irrelevant to coplanarity.

It is worth noting that, apart from making assumptions about the viewpoint of plane observation as in [20], [21], the proposed methods do not require a separate formulation for the coplanar case. This unified representation provides a more compact solution framework.

D. GB Solver Failure Test

When noise is high and the number of points is low, the GB solver may occasionally return obviously incorrect results, likely due to numerical issues. This problem is more severe before applying the weak-perspective approximation. In Fig. 4, we show the number of failures of the GB solver, before and after applying approximation, under 3 points with different noise levels. It shows that the approximation can effectively stabilize the solving process. In practice, if a outlier occurs

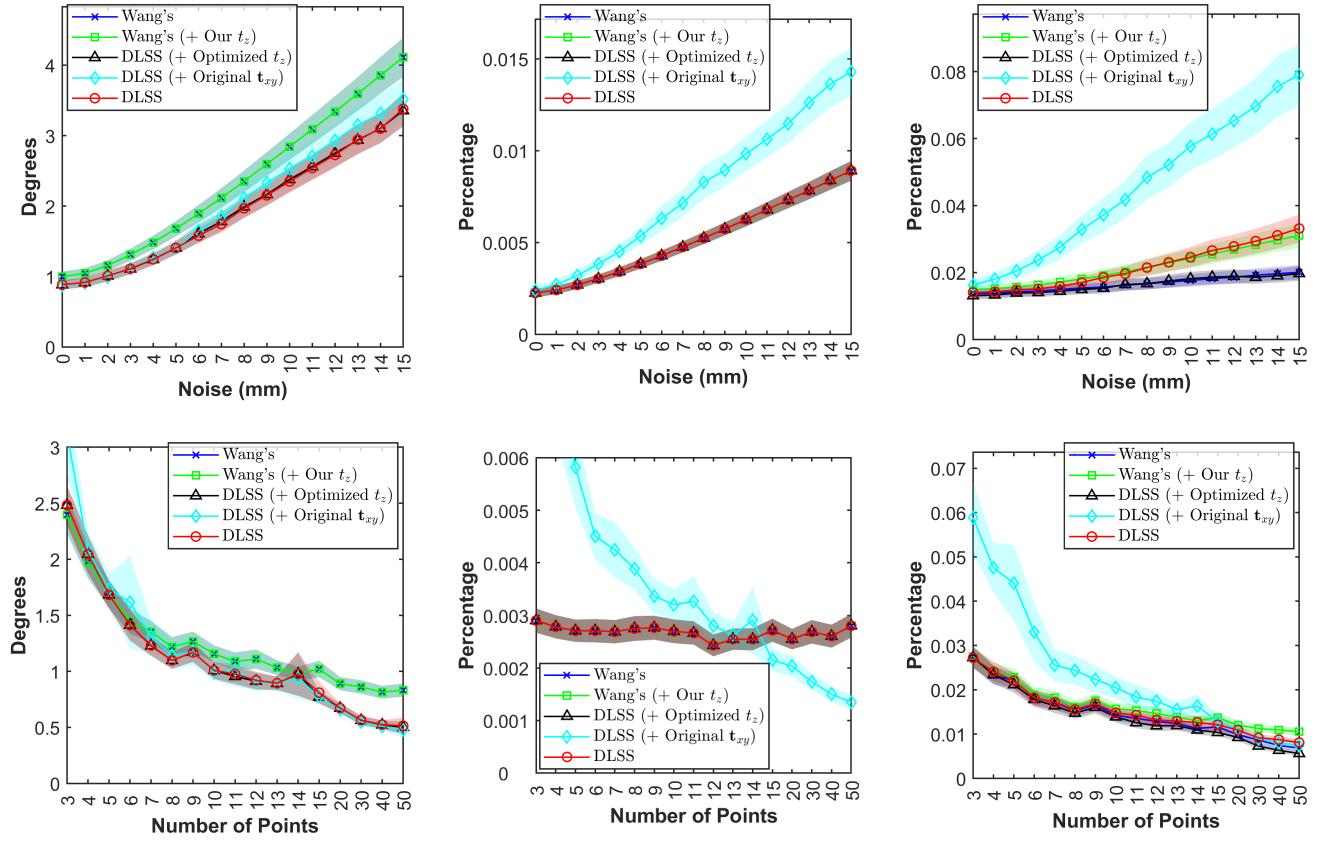


Fig. 3: The results of the coplanar case. The first row represents the results with 10 points under different noise levels. The second row represents the results with 2mm noise under different numbers of points. The first column represents the angular error, the second column represents the t_{xy} error, and the third column represents the t_z error.

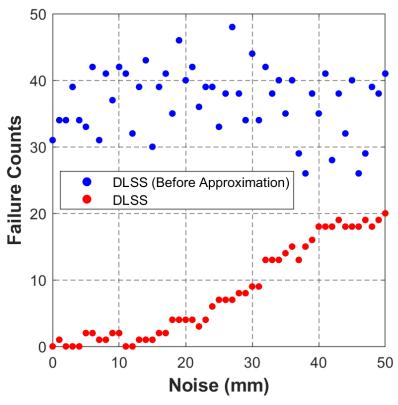


Fig. 4: The results of GB solver failure test.

occasionally, processing the next frame can yield valid results as a workaround.

E. Computation Time

In this section, we first introduce the acceleration strategies applied to DLSS, followed by an analysis of the computation time for different methods and different point configurations.

To begin with, we analyze the time cost composition of

DLSS. While the time cost for running a generated solver is constant, the time required to build the polynomial system depends on the scale of the problem. [32] proposed a vectorization technique to accelerate the construction process. However, due to the different derivation process, the construction of \mathbf{S} cannot benefit from this convenience. The process of obtaining \mathbf{S} through symbolic computation cannot be vectorized, and it is much slower than numerical computation. To accelerate this process, we first use symbolic computation to determine the form of \mathbf{S} for the n -point problem, obtaining a template matrix. Significant speedup can then be achieved through Matlab's just-in-time (JIT) compilation. Another improvement is replacing the **rref** command in the automatically generated solver with **mldivide** as in the implementation of [32]. Despite some loss in precision, using **mldivide** for row reduction can greatly enhance the execution speed.

The time costs of the methods are demonstrated in Fig. 5. The results are derived from the averages of 300 independent trials. Firstly, we observe that DLSS performs similarly in both the general case and the coplanar case. The computation times are approximately 2.5ms and 20ms for 5 points and 30 points, respectively. The slightly increased computation time in the coplanar case is due to the need to verify whether the observed plane's normal satisfies the given constraints. When testing with more than 40 points, the computation time becomes

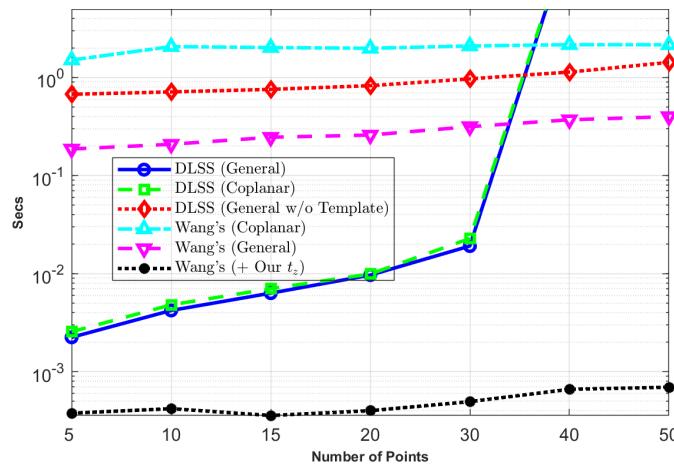


Fig. 5: The time costs of different methods and point configurations. The *y*-axis is presented in log-scale.

unacceptable because the template matrix becomes too large, causing Matlab’s JIT compilation to become extremely slow or even fail. Fortunately, due to the high noise present in low-resolution sonar images, we typically cannot obtain as many points as an optical camera onshore. Additionally, random sample consensus (RANSAC) [6] [14] is commonly used to eliminate the effect of outliers, operating with a fixed number of point pairs, which makes it well-suited to our approach. In the worst case, running the algorithm without a template ensures that it can still function, albeit less efficiently. The time cost for this workaround is shown with a red line in Fig 5. For Wang’s method, the computation time does not vary significantly with the increase in the number of points. In the general case, the time cost for the typical 10 points is approximately 200ms, which is 50 times slower than DLSS. In the coplanar case, due to the need for null space analysis as described in [20], the computation time increases significantly. For Wang’s (+Our t_z), by adopting our proposed closed-form t_z solution, the speed reaches 0.7ms even for 50 points.

Due to differences in computing platforms, programming languages, and specific implementations, the speed of the Wang’s method that we reproduced may not necessarily reflect its efficiency. According to the performance reported in [20], the processing time for fewer than 50 points is approximately 50ms per iteration, which is about 10 times slower than DLSS. As shown in Fig. 5, the significant speed improvements of the proposed algorithm and the closed-form solution for t_z , along with the advantage of providing a unified framework for both general and coplanar cases, are clearly demonstrated.

V. CONCLUSIONS

In this paper, we proposed the DLSS for solving the PnP problem. It achieves higher precision in rotation estimation compared with the previous SOTA, while maintaining comparable performance in translation estimation. The method provides a unified solution framework applicable to both general and coplanar cases, and introduces a closed-form solution for t_z , which significantly accelerates the solving process,

making it 10 times faster than the previous SOTA under common configurations. Simulation experiments validate the effectiveness and efficiency of the DLSS. Future work will focus on investigating the occasional anomalous behavior of the GB solver and exploring other acceleration methods to enhance the scalability. Furthermore, pool tests and sea trials will be conducted to further verify the superiority of DLSS.

REFERENCES

- [1] A. Kim and R. M. Eustice, “Real-time visual slam for autonomous underwater hull inspection using visual saliency,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.
- [2] J. McConnell, I. Collado-Gonzalez, and B. Englöt, “Perception for underwater robots,” *Current Robotics Reports*, vol. 3, no. 4, pp. 177–186, 2022.
- [3] J. Su, H. Li, J. Qian, X. An, F. Qu, and Y. Wei, “A blending method for forward-looking sonar mosaicing handling intra-and inter-frame artifacts,” *Ocean Engineering*, vol. 298, p. 117249, 2024.
- [4] J. Su, J. Qian, X. Tu, F. Qu, and Y. Wei, “Analysis and compensation of acoustic rolling shutter effect of acoustic-lens-based forward-looking sonar,” *IEEE Journal of Oceanic Engineering*, 2024.
- [5] Y. Wang, Y. Ji, H. Woo, Y. Tamura, H. Tsuchiya, A. Yamashita, and H. Asama, “Acoustic camera-based pose graph slam for dense 3-d mapping in underwater environments,” *IEEE Journal of Oceanic Engineering*, vol. 46, no. 3, pp. 829–847, 2020.
- [6] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [7] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [8] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [9] E. Marchand, H. Uchiyama, and F. Spindler, “Pose estimation for augmented reality: a hands-on survey,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.
- [10] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Ep n p: An accurate o (n) solution to the p n p problem,” *International journal of computer vision*, vol. 81, pp. 155–166, 2009.
- [11] J. A. Hesch and S. I. Roumeliotis, “A direct least-squares (dls) method for pnp,” in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 383–390.
- [12] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. Okutomi, “Revisiting the pnp problem: A fast, general and optimal solution,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2344–2351.
- [13] L. Zhou and M. Kaess, “An efficient and accurate algorithm for the perspective-n-point problem,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6245–6252.
- [14] G. Terzakis and M. Lourakis, “A consistently fast and globally optimal solution to the perspective-n-point problem,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 478–494.
- [15] S. Zhuang, Z. Zhao, L. Cao, D. Wang, C. Fu, and K. Du, “A robust and fast method to the perspective-n-point problem for camera pose estimation,” *IEEE Sensors Journal*, vol. 23, no. 11, pp. 11892–11906, 2023.
- [16] S. Negahdaripour, “Calibration of didson forward-scan acoustic video camera,” in *Proceedings of OCEANS 2005 MTS/IEEE*. IEEE, 2005, pp. 1287–1294.
- [17] N. Brahim, D. Guérion, S. Daniel, and B. Solaiman, “3d reconstruction of underwater scenes using didson acoustic sonar image sequences through evolutionary algorithms,” in *OCEANS 2011 IEEE-Spain*. IEEE, 2011, pp. 1–6.
- [18] Y. Wang, Y. Ji, H. Woo, Y. Tamura, H. Tsuchiya, A. Yamashita, and H. Asama, “Planar anp: A solution to acoustic-n-point problem on planar target,” in *Global Oceans 2020: Singapore–US Gulf Coast*. IEEE, 2020, pp. 1–6.

- [19] Y. Wang, Y. Ji, D. Liu, Y. Tamura, H. Tsuchiya, A. Yamashita, and H. Asama, "Acmarker: Acoustic camera-based fiducial marker system in underwater environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5018–5025, 2020.
- [20] Y. Wang, Y. Ji, H. Tsuchiya, J. Ota, H. Asama, and A. Yamashita, "Acoustic-n-point for solving 2d forward looking sonar pose estimation," *IEEE Robotics and Automation Letters*, 2024.
- [21] D. Yang, B. He, M. Zhu, and J. Liu, "An extrinsic calibration method with closed-form solution for underwater opti-acoustic imaging system," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6828–6842, 2020.
- [22] J. Park and J. Kim, "Robust underwater localization using acoustic image alignment for autonomous intervention systems," *IEEE Access*, vol. 10, pp. 58 447–58 457, 2022.
- [23] Y. Wang, Y. Ji, D. Liu, H. Tsuchiya, A. Yamashita, and H. Asama, "Simulator-aided edge-based acoustic camera pose estimation," in *OCEANS 2022-Chennai*. IEEE, 2022, pp. 1–4.
- [24] E. Belcher, W. Hanot, and J. Burch, "Dual-frequency identification sonar (didson)," in *Proceedings of the 2002 international symposium on underwater technology (Cat. No. 02EX556)*. IEEE, 2002, pp. 187–192.
- [25] N. Hurtos, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi, "Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments," *Journal of Field Robotics*, vol. 32, no. 1, pp. 123–151, 2015.
- [26] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [27] H. Stewenius, C. Engels, and D. Nistér, "Recent developments on direct relative orientation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.
- [28] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *Computer Vision-ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III 10*. Springer, 2008, pp. 302–315.
- [29] D. A. Cox, J. Little, and D. O'shea, *Using algebraic geometry*. Springer Science & Business Media, 2005, vol. 185.
- [30] G. Nakano, "Globally optimal dls method for pnp problem with cayley parameterization," in *BMVC*, vol. 78, 2015, pp. 1–78.
- [31] K. Hägglöf, P. O. Lindberg, and L. Svensson, "Computing global minima to polynomial optimization problems using gröbner bases," *Journal of global optimization*, vol. 7, pp. 115–125, 1995.
- [32] Y. Zheng, S. Sugimoto, and M. Okutomi, "Aspnlp: An accurate and scalable solution to the perspective-n-point problem," *IEICE TRANSACTIONS on Information and Systems*, vol. 96, no. 7, pp. 1525–1535, 2013.



Jingyu Qian received the B.S. degree in automation from Hehai University, NanJing, China, in 2019. She received the Ph.D. from the Ocean College, Zhejiang University, Zhoushan, China, in 2024. She is now working as a research assistant in her original research group. Her research interests include underwater acoustic communication, cross-medium communication, marine Internet of Things, and digital signal processing.



Yufan Yuan received the B.E. degree in electronic and information engineering from South China University of Technology, Guangzhou, China, in 2020. She is currently pursuing the Ph.D. degree in ocean engineering and technology with Ocean College, Zhejiang University, Zhoushan, China. Her recent interests include routing protocols and media access control protocols design for underwater acoustic sensor networks.



Fengzhong Qu received the B.S. and M.S. degrees from Zhejiang University, Hangzhou, China, in 2002 and 2005, respectively, both in electrical engineering. He received the Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA, in 2009. From 2009 to 2010, he was an Adjunct Research Scholar in the Department of Electrical and Computer Engineering, University of Florida. Since 2011, he has been with the Ocean College, Zhejiang University, Zhoushan, China, where he is currently a Professor and a Chair of the Institute of Ocean Sensing and Networking. His current research interests include underwater acoustic communication and networks, wireless communications, signal processing, and intelligent transportation systems. Dr. Qu is an Associate Editor of the *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS* and the *IET Communications and China Communications*.



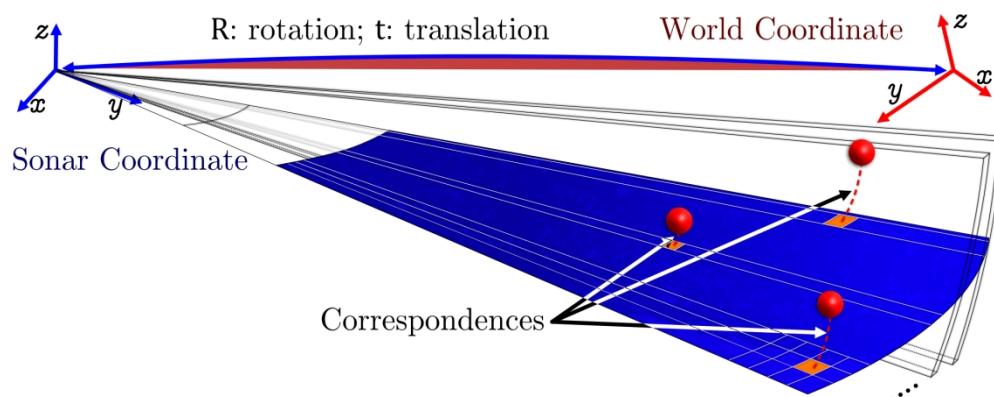
Jiayi Su received the B.S. degree in electronic information engineering from China University of Petroleum, Qingdao, China, in 2019. He is currently pursuing the Ph.D. degree at Ocean College, Zhejiang University, Zhoushan, China. His research interests include underwater computer vision and sonar signal processing.



Haodi Zhu received his B.S. degree in communications engineering from Shandong University, Weihai, China, in 2024. He is currently pursuing his M.S. degree in electronic information with Ocean College, Zhejiang University, Zhoushan, China. His research interests include underwater acoustic sensor networks and artificial intelligence.



Yan Wei received her B.Sc. and M.Sc. degrees from Wuhan University of Technology, Wuhan, China, in 2004 and 2007, respectively, both in naval architecture and ocean engineering. She received her Ph.D. degree from the Department of Maritime and Transportation Technology at Delft University, Delft, the Netherlands, in 2012. Since 2012, she has been with the Ocean College at Zhejiang University and currently is an associate professor. Her recent focus has been the flow noise and its influence in underwater acoustic communications, ocean observatory networking and cross-medium communication.



226x92mm (236 x 236 DPI)