Robot–Camera Calibration in Tightly Constrained Environment Using Interactive Perception

Fangxun Zhong, *Member, IEEE*, Bin Li, *Student Member, IEEE*, Wei Chen, *Student Member, IEEE*, and Yun-Hui Liu, *Fellow, IEEE*

***Abstract*—Manipulation in tight environment is challenging but increasingly common in vision-guided robotic applications. The significantly reduced amount of available feedback (limited visual cues, field of view, robot motion space, etc.) hinders solving the hand-eye relationship accurately. In this article, we propose a new generic approach for online robot–camera calibration that could deal with the least feedback input available in tight environment: an arbitrarily restricted motion space and a single feature point with unknown position for the robot end-effector. We introduce the interactive perception to generate prescribed but tunable robot motions to reveal high-dimensional sensory feedback, which is not obtainable from static images. We then define the interactive feature plane (IFP), whose spatial property corresponds to the robot-actuating trajectories. A depth-free adaptive controller is proposed based on image feedback, where the converged orientation of IFP directly harvests the data for solving the hand–eye relationship. Our algorithm requires neither external calibration sensors/objects nor large-scale data acquisition process. Simulations demonstrate the validity of our method to accurately calibrate different types of robot under various system set-ups. In experiments, we show good results of our algorithm in terms of accuracy and consistency under tight motion space compared to existing approaches using external objects and/or optimization.**

***Index Terms*—Adaptive control, interactive perception (IP), robot–camera calibration, surgical robotics.**

Nomenclature

ζ	Robot's preset target path.
$\mathcal{P}$	Robot's target trajectory upon ζ.
$\mathbf{\Psi}$	Active interaction parameter space.
s	Scalar describing the robot's instant target state.
ψ	Scalar parameter representing the orientation of ζ.
$\mathbf{x}$	Robot's instant state.
$\mathbf{p}_\zeta$	Robot's instant target state induced by $\mathcal{P}$.
$\overline{\mathbf{p}}_\zeta$	Description of $\mathbf{p}_\zeta$ w.r.t. end-effector frame.
$\mathbf{r}/\mathbf{w}$	Robot's (instant) position/orientation.
$\mathbf{r}_\zeta/\mathbf{w}_\zeta$	Robot's (instant) target position/orientation.
$\overline{\mathbf{r}}_\zeta/\overline{\mathbf{w}}_\zeta$	Robot's (instant) local target position/orientation.
$\mathbf{v}$	Vector describing the position of $\mathbf{r}_\zeta$ from $\mathbf{r}_0$.
$\mathbf{m}_\zeta$	Position of the world feature from the robot.
$\mathbf{R}_\zeta$	Rotation matrix from $\mathbf{p}_0$ to $\mathbf{p}_\zeta$.
${}^c\lambda$	Feature depth from the camera.
$\mathbf{y}$	Position of the world feature on 2-D image.
$\mathbf{b}$	vector of the virtual fiducial line on 2-D image.
d	Signed distance of $\mathbf{y}$ w.r.t. $\mathbf{b}$.

Manuscript received 6 May 2023; revised 19 July 2023; accepted 23 July 2023. Date of publication 8 August 2023; date of current version 6 December 2023. This paper was recommended for publication by Associate Editor J. Kelly and Editor S. Behnke upon evaluation of the reviewers' comments. This work was supported in part by the Shenzhen Portion of Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone under Grant HZQB-KCZYB-20200089, in part of the HK RGC under Grant T42-409/18-R and Grant 14202918, in part by the Multi-Scale Medical Robotics Centre, InnoHK, and in part by the VC Fund 4930745 of the CUHK T Stone Robotics Institute. *(Corresponding author: Fangxun Zhong.)*

The authors are with the T Stone Robotics Institute, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, SAR 999077, China (e-mail: fxzhong@cuhk.edu.hk; bli@mae.cuhk.edu.hk; weichen@link.cuhk.edu.hk; yhliu@mae.cuhk.edu.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2023.3299533.

Digital Object Identifier 10.1109/TRO.2023.3299533

I. Introduction

TASK automation entails complex interactions between the robot and working environment. To provide feedback for these interactions, many robots include one or multiple cameras to enable vision-guided operations [1], [2], which requires the spatial transformation between the camera(s) and robot(s). This is regarded as the robot–camera calibration process where the estimated transforms enable the expression of sensor data in a common reference frame.

In most approaches, the model for robot–camera calibration could be mathematically interpreted as solving the AX = XB equation, where acquiring at least two linearly-independent end-effector poses (i.e., varying $\mathbf{A}$ and $\mathbf{B}$) could solve the unknown robot–camera relationship $\mathbf{X}$ [3], [4], [5]. Calibration objects are commonly introduced to generate robust and information-revealing visual measurements. Additional sensors of different modalities, such as optical tracking system [6] and depth sensor [7] could also assist improving calibration accuracy.

However, a significant challenge that many vision-guided robotic operations are commonly facing is the tightly constrained working environment. In robot-assisted minimally invasive surgery [8] or assembly of miniaturized targets [9], for example, the in-vivo environment or narrow pipes/passages are highly enclosed and cluttered. To avoid severe robot downtime, robot–camera calibration should be performed in situ, which prevents external calibration objects or sensors. The field of view of camera is magnified and the end-effector becomes only partially visible [7], [10], while the tight environment renders the data degenerated in parameter space, which hinders the amount of available feedback [11]. Apart from providing a solver, the algorithm should be insensitive to the abovementioned issues as well. Therefore, devising a new robot–camera calibration

approach to systematically address the abovementioned issues becomes crucial for more generic applications.

In this article, we propose a generic approach for robot–camera calibration, which could deal with tightly constrained environments. To adapt to the induced limited visual feedback, we consider the least visual cue possible: a sole salient feature point to be captured by a monocular camera. Neither any calibration objects nor other sensors are available. To complete the calibration process, we leverage the concept of interactive perception (IP) by initiating the robot with prescribed motion patterns to reveal visual feedback with high-dimensional spatial information. We first propose the interactive feature plane (IFP) and its parameter space to fully characterize the spatial property of the robot trajectory. Then, we develop an adaptive depth-free controller fed by the IP-induced 2-D feature error to achieve interactive and stabilized regulation of the IFP parameters from visual guidance. Here, the controller could enforce the robot end-effector motions to a (spherical) space volume of arbitrary size. The controller converges the robot motions to prescribed image features during data collection, despite the unknown 3-D position of the feature. The regulated IFPs through individual trials harvest the 3-D vectors for computing the rotation matrix in a closed-form manner via singular-value decomposition (SVD), which does not require a "good" initial guess. Note that we are unaware of any existing works addressing such constrained environment or using IP to solve this problem. Finally, we validate our framework in both simulations and experiments to show the accuracy, consistency of our algorithm under different environment constraints.

A preliminary study related to this work has been reported in our previous work [12] but only applies to surgical robots, while this work greatly improves the model to make it applicable to any types of 6-DoF robot manipulators. The main contributions of this article include the following.

1) Proposition of the IFP as a novel concept to bridge the robot–camera relationship.
2) A depth-free adaptive controller that online regulates the spatial property of the robot trajectory using image feedback.
3) A closed-form solver of rotation matrix based on regulated IFPs and SVD.
4) Extensive experiments using both simulation and real-word robot platforms to show the applicability of our work.

We also highlight the distinctive advantages of our work in terms of the relaxed assumptions for good practicality.

1) It works in highly restricted motion space, which could be user-specified (a 1-cm-diameter sphere suffices).
2) It uses only a single feature point with unknown 3-D position as the visual cue.
3) No external sensors, calibration objects are required.
4) Neither prior knowledge (e.g., CAD model or the feature position) nor offline training process is required.
5) The procedure is autonomous and extendable to solving eye-in-hand and robot-to-robot calibration problem.
6) Its efficient process ($\sim$1 min) allows online calibration.

The rest of this article is organized as follows. Section II provides the review of related works and summary. Sections III and IV are dedicated to the new robot–camera model of our method using the IFP and the image-guided adaptive controller for data collection. Section V explains the parameter identification for calculating the final calibration result. Sections VI and VII demonstrate the validity of our approach through extensive simulations and experiments, respectively. Section VIII provides the discussions. Finally, Section IX concludes this article.

II. Related Work

Robot–camera calibration for industrial robot manipulators has been actively studied through decades under a variety of applications [13]. Typical steps include modeling, measurement, identification, and compensation. One popular framework is to use a planar calibration object to derive 3-D information from single images. It is adopted by classic works including [3], [14], [15], [16] that solve the $\mathbf{AX} = \mathbf{XB}$ with linear solutions upon either single-stage or dual-stage estimation. Daniilidis et al. [17] introduced dual quaternion representation to simultaneously solve rotation and translation to avoid propagation error. These works provide analytical solvers to recover the transformation matrix. There are also works that introduce iterative approaches using linear matrix inequality [18], or nonlinear optimization [19], [20], [21]. Zhao [22] further formulated the calibration problem into convex optimization where a global solution could be obtained. There are also works in [13] and [23] that characterize the problem using $\mathbf{AX} = \mathbf{ZB}$ and solve it by iterative methods. In [24], different types of calibration objects have been used for providing visual measurements. Researchers in [25] and [26] further develop data selection policy and report its superior performance compared to manual and/or random selection. Maye et al. [11] further introduced an information-theoretic method to optimize data collection sequence to achieve improved accuracy of online calibration. Note that all the abovementioned works require the camera to capture preset calibration objects and could only be performed offline, which is also the case for robot-to-robot calibration in [27] and [28]. The dependence of calibration objects also implies a free workspace to generate enough spatial disparity in data collection, which is unfriendly to spatially confined environment but has not been evaluated by existing works. More comprehensive reviews of existing robot–camera calibration approaches could be referred to [1], [29].

Recently, many works aim to tackle robot–camera calibration without using calibration objects. For example, the works in [11], [30], [31], [32], and [33] suggested analyzing the visual rigidity of the world scene during data collection to establish constraints for parameter estimation without using known patterns. However, this requires substantial features within the visual environment and is not guaranteed under limit field of view. Koide et al. [34] approached hand-eye calibration by directly minimizing the reprojection error of an observed image, which does not have to contain specific patterns. However, a known planar image is still required to generate back-projection errors. Moreover, efforts have been made to tackle robot–camera calibration based on vision-based pose estimation of the observed

3-D objects [35], where the pose estimation process requires a preset database to the target objects, which involves considerable offline workload. Gratal et al. [36] and Hu et al. [37] introduced the LED and line laser, respectively, to generate light pattern for calibration instead of physical objects.

Online robot–camera calibration in tight environment is particular required by surgical robots in surgeries. To tackle such issues, Ye et al. [38] and Allan et al. proposed iterative estimation of the eye-to-hand relationship based on back-projection errors to estimate the pose of the robotic surgical instruments. However, both algorithms require reasonable initialization and intensive computation (either costing $\sim$1 s or requiring a workstation-level PC for real-time computation). The works in [39] and [40] utilized remote center-of-motion (RCM) kinematic constraints of the surgical robot such that natural appearance of the robot could be directly used as visual measurements. However, the calibration accuracy does not suffice robot automation of delicate surgical procedures ($>$ 10 mm in translation error). In [41], a 2.0 mm robot positioning accuracy is achieved, but relied on both the external calibration object and optical tracking system. Roberti et al. [42] further performed robot–camera calibration by introducing the RGB-D sensor, which is difficult to be integrated in the confined space in minimally invasive set-ups. Richter et al. [10] considered the (internal) kinematic sensing errors from the joint data by using a new parameter for evaluation. Experimental validations are conducted on different types of robots. However, the algorithm depends on multiple preset visual features on the robot end-effector, which need to be tracked by a trained learning algorithm.

III. Problem Formulation

Our approach targets a general 6-DoF serial robot manipulator whose end-effector could perform motion in full SE(3) space. This indicates that mainstream manipulators including 6-DoF PUMA-like robots, SCARA-like robots, and articulated robots, such as the da Vinci surgical system (dVSS), and RAVEN robot, are all qualified. It could also be applied to any redundant robot manipulators with N DoFs ($N > 6$), as long as it owns SE(3) workspace. A precalibrated monocular camera is used to acquire visual feedback, whose configuration regarding the robot could be either eye-to-hand or eye-in-hand in our calibration (eye-to-hand as in our validation). The abovementioned set-up is commonly adopted by many robot applications.

As mentioned, tight environment brings two primary constraints to solving robot–camera calibration: the limited robot motion space and the limited amount of visual feedback available. Without loss of generality, we assume the simplest form of visual cue: a single salient world point fixed on the end-effector (i.e., drivable by the last robot joint), whose relative position from the end-effector should be unknown as well. This suggests that such point could be selected based on any features available on the robot structure, e.g., a bolt endpoint, a manually labeled dot, or even a firm stain. Note that, these have been rarely explored by existing approaches, but practically demanded by tasks such as inspection of pipes or passages, or robotic surgery in in-vivo environment. They mutually own a highly concentrated operating space plus the highly limited field of view of the camera, where only a part of end-effector is visually observable. Then, the ultimate goal is to complete the robot–camera transformation by solving the following well-adopted term:

$$\mathbf{T}_c = \begin{bmatrix} \mathbf{R}_c & \mathbf{t}_c \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{1}$$

where $\mathbf{R}_c \in \mathbb{R}^{3\times 3}$ and $\mathbf{t} \in \mathbb{R}^3_c$ denote the rotation matrix and translation vector. The proposed approach yielding the abovementioned formulations should also be easily extended to data-sufficient scenarios, e.g., multiple feature points, line segments, or multicamera/-robot set-ups.

In this article, we use the italic lowercase letters or Greek alphabets to denote scalars s. A vector and a matrix will be denoted by bold lowercase letter or Greek alphabet and bold capital letter, respectively, ($\mathbf{x}/\boldsymbol{\eta}$ as vectors, $\mathbf{T}$ as a matrix). Particularly, the list of defined variables in our modeling to be appeared in this article is shown in the Nomenclature for reference.

IV. Interactive Perception

IP, apart from the concept of active perception, aims to introduce deliberate actions to the robot manipulating target(s) whose motions are online reactive to corresponding sensor-based measurements. This could reveal additional sensory feedback, which is otherwise unavailable if we merely record the instant input–output data. The concept of IP has been systematically addressed in the work [43] and has been widely applied to robot manipulation tasks involving physical interactions to the environment [44], [45]. As we only rely on a single feature point in this work, IP is potentially powerful to increase the dimension of image feedback to 2-D or 3-D geometry. In the following part of this section, we will propose an IP model with its control strategy to solve robot–camera calibration.

A. Robot Actuation Model

First, we define $\mathcal{P}(\zeta, \cdot)$ as a predefined and online-tunable spatial trajectory with its geometric path denoted by ζ to initiate IP, active interaction space and its parameters

$$\boldsymbol{\Psi} = \begin{bmatrix} s(t) & \psi(t) & \phi \end{bmatrix}^{\mathsf{T}} \tag{2}$$

where $s(t)$ is the bounded and C^2-continuous timing function that parametrizes the evolution of $\mathbf{p}_\zeta(\cdot)$ to determine the trajectory $\mathcal{P}(\zeta, \boldsymbol{\Psi}(t))$. $\phi \in \mathbb{R}$ is the motion space constraint that rigorously restricts the robot actions subject to ζ within a spherical space (ϕ being the radius) throughout data acquisition. The sphere is assigned relative to the initial position of the robot end-effector, and ϕ could be arbitrarily selected as long as meeting the tight environment. ψ is the scalar that describes the spatial orientation of $\mathcal{P}$. During robot–camera calibration, $\mathcal{P}$ should be ideally fed to the output space of the robot's end-effector in the sense that

$$\mathcal{P} = \left\{ \mathbf{x} \in \mathbb{R}^6 \mid s(t) \in [s_s, s_f], \psi(t), \phi(t) \mapsto \mathbf{x} = \mathbf{p}_\zeta(\boldsymbol{\Psi}(t)) \right\} \tag{3}$$

where $\mathbf{x}$ denotes the state of the robot described in Cartesian space, and $\mathbf{p}_\zeta(\cdot) \in \zeta$ is the instant path descriptor varied by $s(t)$. To facilitate IP to provide useful sensor information for robot–camera calibration under data-deficient scenarios, the property of $\mathcal{P}$ should satisfy certain constraints to generate new valid visual feedback propagated from a single point. Here, we propose that a path ζ could be qualified as a candidate IP path if it simultaneously satisfy the following properties.

1) $\mathcal{P}$ is of C^1 continuity subject to a second-order differentiable $s(t)$.
2) The path ζ of $\mathcal{P}$ is planar in 3-D Cartesian space given any constant ψ.
3) The path ζ of $\mathcal{P}$ is convex.

Now, we start the robot model by giving the notation of the robot kinematics $\mathbf{x}$ as follows:

$$\mathbf{x} = \begin{bmatrix}\mathbf{r}(\mathbf{q}(t)) & \mathbf{w}(\mathbf{q}(t))\end{bmatrix} = \mathbf{f}(\mathbf{q}(t)) \tag{4}$$

where $\mathbf{r}(\mathbf{q}(t)) \in \mathbb{R}^3$ and $\mathbf{w}(\mathbf{q}(t)) \in \mathbb{R}^3 \subseteq \mathrm{so}(3)$ denote the position and orientation of the robot's end-effector, respectively. $\mathbf{q}(t)$ indicates the generalized joint state of the robot. Then, one can compute an instant ideal robot state relative to a known initial joint state $\mathbf{q}_0$ as follows:

$$\mathbf{p}_{\zeta,\mathbf{q}_0}(\mathbf{\Psi}(t)) = \mathbf{x}_{\mathbf{q}_0} + \begin{bmatrix}\mathbf{R}_{\mathbf{q}_0} \cdot \underbrace{\phi(t)\mathbf{R}^\dagger(\psi(t))\bar{\mathbf{r}}_\zeta(s(t))}_{\mathbf{v}} \\ \overline{\mathbf{w}}_\zeta(t)\end{bmatrix} \tag{5}$$

where $\bar{\mathbf{r}}_\zeta(\cdot)$ denotes the path descriptor relative to the local frame, $\overline{\mathbf{w}}_\zeta(t)$ is the orientation along the path to be assigned. $\mathbf{v}(\cdot) \in \mathbb{R}^3$ is the 3-D vector that calculates the instant target point of trajectory $\mathcal{P}$ over time with respect to $\mathbf{x}_{\mathbf{q}_0}$. $\mathbf{x}_{\mathbf{q}_0}$ denotes the robot state computed by $\mathbf{q}_0$. Here, we assign a simple user-defined form of $\mathbf{R}^\dagger(\psi(t))$ as[1]

$$\mathbf{R}^\dagger(\psi(t)) = \begin{bmatrix}1 & 0 & 0 \\ 0 & \cos\psi(t) & -\sin\psi(t) \\ 0 & \sin\psi(t) & \cos\psi(t)\end{bmatrix}. \tag{6}$$

From (5), the target trajectory is set to be described based on the robot's initial state, allowing on-demand recalibration upon a given instant configuration during a task, where a suitable workspace could be arbitrarily selected. Here, we name the 3-D plane that contains the path ζ as the IFP. Its orientation is then defined by $\mathbf{\Psi}(t)$, which is to be online interacted to visual feedback and plays a crucial role to bridge robot–camera relationship.

Consider a (sole) world feature point $\mathbf{m} \in \mathbb{R}^3$ mounted on the end-effector. In order to achieve image-guided IP, the prescribed path ζ generated by (5) is propagated to the feature point, which further suggests

$$\mathbf{m}_{\zeta,\mathbf{q}_0}(\mathbf{\Psi}(t)) = \mathbf{r}_{\zeta,\mathbf{q}_0}(\mathbf{\Psi}(t)) + \mathbf{R}_{\mathbf{q}_0}\mathbf{R}_\zeta(\overline{\mathbf{w}}_\zeta)\bar{\mathbf{t}}_m \tag{7}$$

where $\mathbf{m}_{\zeta,\mathbf{q}_0}(\cdot)$ is the target 3-D position of the feature to be guided by the robot while deploying IP, $\mathbf{r}_{\zeta,\mathbf{q}_0}(\cdot) \in \mathbb{R}^3$ is the position part of $\mathbf{p}_{\zeta,\mathbf{q}_0}(\cdot)$, $\mathbf{R}_\zeta(\overline{\mathbf{w}}_\zeta)$ denotes the composite rotation matrix computed from the user-defined orientation $\overline{\mathbf{w}}_\zeta$ in Euler-angle form. The term $\bar{\mathbf{t}}_m \in \mathbb{R}^3$ represents the unknown but constant position of the feature $\mathbf{m}$ with respect to the robot end-effector (geometric interpretation as shown in Fig. 1).

[1]The form of $\mathbf{R}^\dagger$ does not have to be complex, as long as the varying $\psi(t)$ could lead to orientation change of $\mathbf{v}$ to generate effective image feedback.

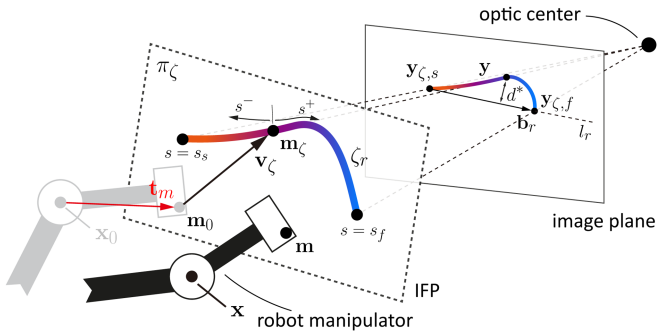


Fig. 1. Geometric interpretation of the robot–camera interaction model.

B. Visual Feedback Model

Given the interaction input provided by the robot, we then need to correlate the robot actions to the feature despite the unknown information of $\mathbf{m}$ from the robot end-effector and the robot–camera transformation matrix. Consider $\mathbf{m}$ to be observable by the camera as $\mathbf{y} \in \mathbb{R}^2$ on the image. Its possible 3-D position with respect to the camera is computed as follows:

$$\begin{bmatrix}\mathbf{y}^\intercal(\cdot) & 1\end{bmatrix}^\intercal = {}^c\lambda(\mathbf{m}_{\zeta,\mathbf{q}_0}(\cdot))\mathbf{K}\mathbf{T}_c^{-1}(\mathbf{r}_{\zeta,\mathbf{q}_0}(\mathbf{\Psi}(t))) + \bar{\mathbf{t}}_m) \tag{8}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the intrinsic matrix of the camera upon pinhole projection model,[2] ${}^c\lambda \in \mathbb{R}$ is the parameter that contains unknown depth information of $\mathbf{m}$ subject to $\mathbf{T}_c$ and $\bar{\mathbf{t}}_m$ in a nonlinear expression. Thus, unlike calibration using external objects, direct recovery of 3-D sensory feedback by solely inspecting static $\mathbf{y}$ through individual images now becomes impossible. To tackle this issue, additional visual reference should be introduced to extend the dimension of the feedback that interprets the relationship between the robot and the camera. As basis of previous modeling, we define a virtual fiducial line (VFL), which contains a 3-D line segment in which the robot motion stays at the start of the calibration process. Its appearance on the 2-D image denoted by l_r is further parametrized by the following vector:

$$\mathbf{b}_r = \mathbf{y}_{\zeta,f}(\cdot) - \mathbf{y}_{\zeta,s}(\cdot) \tag{9}$$

where $\mathbf{y}_{\zeta,s}$ and $\mathbf{y}_{\zeta,f}$ denote the projection of $\mathbf{m}$ evaluated at $s = s_s$ and $s = s_f$, respectively, which are inherently available while the robot performs trajectory tracking guided by $\mathbf{p}_\zeta(\cdot)$. Then, we could interpret the scaled "point-to-line" distance between the VFL l_r and the projected feature $\mathbf{y}$

$$d(\mathbf{y}) = \mathbf{b}(\mathbf{y})^\intercal\mathbf{b}_{r_\perp} \tag{10}$$

where $\mathbf{b}_{r_\perp}$ denotes the (anticlockwise) perpendicular vector of $\mathbf{b}_r$, $\mathbf{b}(\mathbf{y}) = \mathbf{y} - \mathbf{y}_{\zeta,s}$, which varies over time subject to robot

[2]As we have assumed calibrated camera intrinsics, the distortion of the captured image for detecting the feature point has been identified and rectified in advance for the later data acquisition.

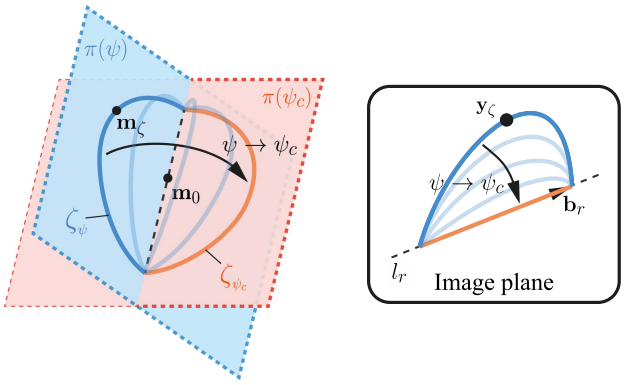


Fig. 2. Illustration of the interactive control process, where (a) orientation regulation of the IFP in 3-D space, and (b) corresponding convergence of the feature $\mathbf{y}$ to the VFL l_r as the robot control input.

actuation. Note that, d represents the signed distance between the feature point and the VRL without normalization. This is to avoid abrupt change of d while its norm approaches 0. Additionally, the sign of $d(\mathbf{y}) \in \mathbb{R}$ subject to the robot-actuating feature $\mathbf{m}$ is consistent for all $s \in [s_s, s_f]$ given a convex path ζ with respect to the VFL l_r in (7). As the curve of path ζ is convex, its enclosed area with l_r is still convex [46] in the 2-D image. This suggests the projection of ζ always stays within one side of l_r except the interception points, i.e., when $s = s_s$ and $s = s_f$ (refer to Fig. 1 for illustration).

C. Adaptive Interaction Control

Now, we aim to devise an robot control strategy such that the property of the IP-relevant motions described by $\mathbf{\Psi}$ is tunable by the visual feedback model despite the presence of unknown $\mathbf{T}$ and $\bar{\mathbf{t}}_m$. Here, as ζ is assumed a planar path itself in Section IV-A, we expect the IP to online tune the orientation of the path (via ψ) based on image feedback, until the projected trajectory of the feature point $\mathbf{m}$ on the image stays within the projected 2-D line of l_r. We will also show this leads ψ converged to a constant value, namely ψ_c (see Fig. 2 as conceptual demonstration of the expected control process). To start with, we differentiate the model of the VRL l_r such that

$$\mathbf{b}_{r_\perp}^\intercal \dot{\mathbf{b}}(\mathbf{y}) = \boldsymbol{\eta}(\mathbf{m}_\zeta)\dot{\mathbf{m}}_\zeta \tag{11}$$

where $\boldsymbol{\eta}(\cdot) : \mathbb{R}^3 \to \mathbb{R}$ maps the linear velocity of the prescribed feature motion $\dot{\mathbf{m}}$ to its resultant change rate of the scaled distance d on the image, with

$$\boldsymbol{\eta}(\cdot) = \frac{{}^c\lambda(\mathbf{m}_\zeta) - \mathbf{m}_\zeta \mathbf{r}_3^\intercal}{{}^c\lambda^2(\mathbf{m}_\zeta)} \mathbf{b}_{r_\perp}^\intercal \mathbf{K} \mathbf{T}_c^{-1} \tag{12}$$

where $\mathbf{r}_3$ is the third row in $\mathbf{R} \in \mathbb{R}^{3\times3}$, which is unknown during IP. Meanwhile, taking the derivative of (7) and substitute to (11) could further lead to the following relationship:

$$\dot{d} = \boldsymbol{\eta}(\mathbf{m}_\zeta) \left(\frac{\partial \mathbf{r}_{\zeta,\mathbf{q}_0}}{\partial \mathbf{\Psi}}(\mathbf{\Psi}(t))\dot{\mathbf{\Psi}} + \mathbf{R}_{\mathbf{q}_0} \dot{\mathbf{R}}_\zeta(\mathbf{w}_\zeta(t)) \bar{\mathbf{t}}_m \right). \tag{13}$$

Note that, by inspecting (13), $\dot{d}(t)$ contains the unknown term $\bar{\mathbf{t}}_m$, which is coupled with the unknown constants in $\boldsymbol{\eta}(\cdot)$ and might not be accurately identified. This will render the robot-enabled state $\mathbf{w}(t)$ to be uncontrollable. However, as the property of the path ζ could be manually selected, such issue is solvable by enforcing $\mathbf{w}(\mathbf{t})_\zeta \equiv \mathbf{0}_{3\times1}$ such that $\dot{\mathbf{R}}_\zeta(\mathbf{w}_\zeta(t)) = 0$. This physically means that, we enforce the orientation of the robot end-effector upon ζ to be identical to $\mathbf{w}_0$ while the robot is tracking the trajectory. This further indicates that the orientation of the end-effector $\mathbf{w}_\zeta(t)$ throughout the data collection part remains identical to $\mathbf{w}_0$ in $\mathbf{x}_0$. This is necessary to provide a convergable data input in VFL while the position of $\mathbf{m}$ on the end-effector is unknown.

As stated, the calibration is to be conducted in confined environment where the robot workspace should be restricted, we could simply select a constant motion magnitude ϕ such that $\dot{\phi} \equiv 0$. Meanwhile, as the timing function $s(t)$ in $\mathbf{\Psi}$ is preset, we could further arrange (13) into a mapping independent of only controlling variables

$$\dot{d}(\cdot) = \underbrace{\phi \boldsymbol{\eta}(\mathbf{m}_\zeta) \frac{\partial \mathbf{r}_{\zeta,\mathbf{q}_0}}{\partial \mathbf{\Psi}}(\mathbf{\Psi}(t)) \mathbf{H}(\dot{s}(t))}_{\mathbf{A}(\cdot)} \mathbf{h}(\dot{\psi}(t)) \tag{14}$$

where

$$\mathbf{H}(\dot{s}) = \begin{bmatrix} \dot{s} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \mathbf{h}(\dot{\psi}) = \begin{bmatrix} 1 \\ \dot{\psi} \end{bmatrix} \tag{15}$$

leaving the path orientation ψ being the final tunable variable during IP. The equation in (14) states an important relationship about how the variation of the trajectory (enabled by s and path property regulated by ψ and ϕ) could affect the change of image-based feedback via online measurement of the sole feature $\mathbf{y}$. It allows us to perform active regulation of interaction space parameters $\mathbf{\Psi}$ until the image feedback satisfies a condition that implies robot–camera relationship.

Now, we need to introduce a parameter regulator to determine the dynamics of ψ such that the IP-induced motion could not only generate useful visual feedback of d but also regulate the property of the IFP containing the trajectory $\mathcal{P}$ via (14). This should be achieved with the presence of unknown parameters. By inspecting (12), it is noticeable that $\mathbf{T}_c$ and the elements of $\bar{\mathbf{t}}_m$ solely appear as the factored form in $\boldsymbol{\eta}(\cdot)$. Thus, the mapping in (11) could be algebraically arranged into

$$\dot{d}(\cdot) = \mathbf{W}(\mathbf{r}_\zeta(t), \mathbf{\Psi}(t), \dot{\mathbf{\Psi}}(t))\boldsymbol{\theta} \tag{16}$$

where $\mathbf{W}(\cdot) \in \mathbb{R}^{1\times1}$ is the regressor matrix constructed by known or measurable variables. Note that $\boldsymbol{\theta} \in \mathbb{R}^l$ is a column vector from which the elements are computed as a composite form of $\bar{\mathbf{t}}_m$ and $\mathbf{T}_c$. As we assume to know no prior knowledge of them, we must estimate $\boldsymbol{\theta}$ online instead of $\bar{\mathbf{t}}_m$ and $\mathbf{T}_c$ while reacting to the visual feedback d. Denote an estimation of the unknown parameter vector by $\hat{\boldsymbol{\theta}}$, then we could define the following error vector flow:

$$e_d = \mathbf{W}(\cdot)(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) = \mathbf{W}\Delta\boldsymbol{\theta} \tag{17}$$

where the error $e_d \in \mathbb{R}^2$ is continuously measurable from 2-D image, which is explicitly attributed to the estimation error $\boldsymbol{\Delta\theta}$. By proposing the following update rule:

$$\frac{d}{dt}\hat{\boldsymbol{\theta}} = -\kappa \mathbf{W}(\cdot)^{\intercal} e_d \tag{18}$$

could lead the feedback flow of e_d to be stably converged to zero, with $\kappa \in \mathbb{R}$ is the positive-definite tuning matrix for the update steps. Moreover, the updating rule in (18) results in stable convergence of $\boldsymbol{\theta}$. This could be proved by initiating a Lyapunov function in quadratic form $\{V \in \mathbb{R} \mid V = \frac{1}{2}\Delta\boldsymbol{\theta}^{\intercal}\Delta\boldsymbol{\theta}\}$. Then, differentiating V combining with (17) could lead to $\dot{V} = -e_d\kappa^2\mathbf{W}(\cdot)\mathbf{W}(\cdot)^{\intercal}e_d \leq 0$, which is passive and Lyapunov-stable [47]. However, the online estimation process of $\boldsymbol{\theta}$ that contributes to minimization of e_d does not necessarily indicate an accurate recovery of the elements value θ, as the main purpose of such adaptive estimation is to enable convergent performance of the online regulation of ψ subject to IP-induced robot motions. Thus, we propose a new regulator to independently determine the dynamics of d, which is reactive to the visual feedback d

$$\mathbf{h}(\dot{\psi}) = -\frac{\gamma}{||r_1||}\mathbf{A}^{+}(\mathbf{m}_{\zeta}, \boldsymbol{\Psi}(t))d(t) \tag{19}$$

where $||r_1|| > 0$ normalizes the first component of the subsequent term to enforce the homogeneous form of $\mathbf{h}(\dot{\psi})$ in (15), $\gamma \in \mathbb{R}$ is the constant positive gain, $\mathbf{A}^{+}(\cdot) \in \mathbb{R}^{2\times 1}$ denotes the pseudoinverse of $\mathbf{A}(\cdot)$.

Theorem 1: The regulator of $\mathbf{h}(\dot{\psi})$ in (19) stably tunes the orientation of the IFP, which eliminates the distance between $\mathbf{y}$ and the VRL as $\mathbf{b}_r$, i.e.:

$$\lim_{t\to\infty} ||d(t)|| = 0. \tag{20}$$

Proof: Consider the following dynamics of d subject to (11) fed by the above regulator:

$$\dot{d}(t) = -\frac{\gamma}{||r_1||}\mathbf{A}(\cdot)\mathbf{A}^{+}(\cdot)d(t) \tag{21}$$

where the factored term of $d(t)$ is obviously a negative real scalar. Thus, the dynamics in (21) is passive, which decreases $d(t)$ to zero.

Up to now, the vision-guided IP model between the robot motion trajectory and its visual feedback is completed. We have initiated a robot-enabled trajectory $\mathcal{P}$ with its path ζ_r and evolution $s(t)$, and then utilize (19) to achieve interactive regulation of the orientation of the path ζ_r based on visual feedback $d(t)$. The regulation of ψ could stably minimize $d(t)$ to zero via adaptive estimation of (18) regardless of the unknown depth of the feature $\mathbf{m}$ with respect to the camera. Before we proceed to how the dynamics (21) contributes to the robot–camera calibration, a controller must be designed to guide the robot to follow the time-varying trajectory $\mathcal{P}$ such that the above IP-enabled motions could be tracked. Note that, from (7), the robot-actuated motion $\mathbf{x}(\mathbf{q})$ is further determined by the world feature point $\mathbf{m}_{\zeta,\mathbf{q}_0}(\boldsymbol{\Psi}(t))$ to achieve vision-guided IP. Here, we propose the following robot controller:

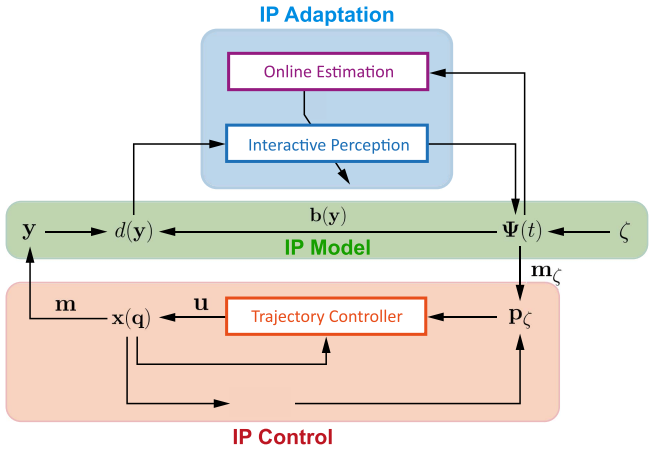


Fig. 3. Overall diagram of IP for data collection.

$$\begin{aligned}\mathbf{u} = &-\mathbf{J}^{-1}(\mathbf{q}(t))\boldsymbol{\Gamma}\begin{bmatrix}(\mathbf{r}(t) - \mathbf{x}_{\mathbf{q}_0} - \phi\mathbf{R}_{\mathbf{q}_0}\mathbf{v}(\boldsymbol{\Psi}(t))) \\ \mathbf{0}_{3\times 6}\end{bmatrix} \\ &-\phi\mathbf{J}^{-1}(\mathbf{q}(t))\begin{bmatrix}\mathbf{R}_{\mathbf{q}_0}\frac{\partial \mathbf{v}}{\partial \boldsymbol{\Psi}}(\boldsymbol{\Psi}(t))\dot{\boldsymbol{\Psi}}(t) \\ \mathbf{0}_{3\times 6}\end{bmatrix}\end{aligned} \tag{22}$$

where $\mathbf{J}(\mathbf{q}(t)) \in \mathbb{R}^{6\times 6}$ denotes the Jacobian matrix from (4), $\boldsymbol{\Gamma} \in \mathbf{R}^{6\times 6}$ is the positive-definite gain matrix, $\dot{\psi}$ corresponds to the first element of the normalized term in $\mathbf{h}(\dot{\psi})$. Meanwhile, if we define the control input as $\mathbf{u}(t) = \dot{\mathbf{q}}(t)$, the controller (22) leads to the following theorem.

Theorem 2: The robot motion initiated by (22) guarantees stable tracking of the robot-mounted feature point $\mathbf{m}(\mathbf{q}(t))$ to its IP-induced trajectory as $\mathbf{m}_{\zeta,\mathbf{q}_0}(\boldsymbol{\Psi}(t))$, i.e.:

$$\lim_{t\to\infty} ||\mathbf{m}(\mathbf{q}(t)) - \mathbf{m}_{\zeta,\mathbf{q}_0}(\boldsymbol{\Psi}(t))|| = 0. \tag{23}$$

Proof: See Appendix A.

Meanwhile, the orientation term $\mathbf{w}_{\zeta}$ in (36) could also be tracked as $\dot{\mathbf{w}} - \dot{\mathbf{w}}_{\zeta} + \boldsymbol{\Gamma}(\mathbf{w} - \mathbf{w}_{\zeta}) = \mathbf{0}_{3\times 1}$ whose asymptotic stability to $\mathbf{w} = \mathbf{w}_{\zeta}$ is guaranteed, as we have assigned an unchanged orientation of $\mathbf{R}_{\mathbf{q}_0}$, with $\dot{\mathbf{w}}(\mathbf{q}(t)) = \mathbf{0}_{3\times 1}$. This indicates that the controller $\mathbf{u}$ deployed to the robot joint velocities could lead to the motion of the feature point $\mathbf{m}$ to stably track the prescribed IP-relevant trajectory $\mathbf{m}_{\zeta}$ despite the unknown $\mathbf{t}_m$ (achieved by $\mathbf{w}_{\zeta}(t) = \mathbf{w}_0$). In addition, the motion contributes to an adaptive image-guided feedback convergence of $d(t)$ to 0 subject to online tuning of ψ. The overall IP-based data acquisition process is referred to Fig. 3 . The IP model not only facilitates image-based robot control but is also able to deal with the unknown $\mathbf{t}_m$ during orientation regulation of IFP. In the following section, we will elaborate how the collected data is interpreted to solve the robot–camera transformation matrix $\mathbf{T}_c$.

V. Identification

A. Data Interpretation

Now, we seek to recover the spatial relationship between the robot and the camera using the results harvested by IP-based motions. Consider a settled $d(t)$ to 0 subject to the controller $\mathbf{u}$ in (22). By inspecting the mapping (14), the change of $\dot{\psi}(t)$, or the orientation of the IP-induced path ζ_r, is minimized to 0 as well. Thus, the following implication holds:

$$\psi(t) \to \psi_c \iff d(t), \dot{d}(t) \to 0 \tag{24}$$

where ψ_c is a constant under a given $\mathbf{x}_{\mathbf{q}_0}$ and the property of ζ_r. This implies that the orientation of the IFP π_r is also settled, whose normal vector could be computed as follows:

$$\begin{aligned} \overline{\mathbf{n}}_\zeta(\phi, \psi_c) &= \frac{\mathbf{v}(\phi, \psi_c, s_1) \times \mathbf{v}(\phi, \psi_c, s_2)}{||\mathbf{v}(\phi, \psi_c, s_1)||\,||\mathbf{v}(\phi, \psi_c, s_2)||} \\ \text{s.t.} \quad & s_1, s_2 \in [s_s, s_f], s_1 < s_2 \end{aligned} \tag{25}$$

from which one could further transform it to the robot base as $\mathbf{n}_{\pi,\zeta} = \mathbf{R}_{\mathbf{q}_0}\overline{\mathbf{n}}_\zeta(\phi, \psi_c)$. Meanwhile, as d stably converges to 0, the projection of the IFP subject to path ζ_r on the image is exactly the 2-D line described by $\mathbf{y}_{\zeta,s}$ and $\mathbf{y}_{\zeta,f}$. Denote their 3-D positions with respect to the camera (on the focal plane) as $\mathbf{c}_s \in \mathbb{R}^3$ and $\mathbf{c}_f \in \mathbb{R}^3$, they could be computed from the camera intrinsic parameters as

$$\mathbf{c}_s = f\mathbf{K}^{-1}\begin{bmatrix}\mathbf{y}_{\zeta,s}^\intercal & 1\end{bmatrix}^\intercal, \quad \mathbf{c}_f = f\mathbf{K}^{-1}\begin{bmatrix}\mathbf{y}_{\zeta,f}^\intercal & 1\end{bmatrix}^\intercal \tag{26}$$

where $f \in \mathbb{R}$ is the focal length. Thus, the orientation of the IFP under the settled ψ_c is also computable

$${}^c\mathbf{n}_\zeta = \frac{\mathbf{c}_s \times \mathbf{c}_f}{||\mathbf{c}_s \times \mathbf{c}_f||} \tag{27}$$

with ${}^c\mathbf{n}_\zeta \in \mathbb{R}^3$ denotes the identical IFP represented relative to the camera. This states an important point that the IFP enables the feedback interpretation from $d(t)$ to expand the dimension of the accessible data from 2-D (a single feature point $\mathbf{m}$) to 3-D (IFP normal vectors) by using IP. This is otherwise unavailable to static, noninteraction image feedback where planar patterns must be introduced, which normally appears in existing robot–camera calibration approaches.

B. Parameter Solver

At this stage, we elaborate how to recover the robot–camera transformation matrix based on the abovementioned results. Without loss of generality, consider a 3-D unit-length vector $\mathbf{e} \in \mathbb{R}^3$ described by two unparallel Cartesian coordinate frames $\mathcal{F}_1$ and $\mathcal{F}_2$ as ${}^1\mathbf{e}$ and ${}^2\mathbf{e}$, respectively. Then, the rotation matrix (notated as ${}^1\mathbf{R}_2 \in \mathbb{R}^3$) between such two frames also indicates a single-step rotation operation that reorients $\mathbf{e}_2$ to become parallel to $\mathbf{e}_1$ (or the other way around). If we describe ${}^1\mathbf{R}_2$ via axis-angle representation with $\mathbf{a} \in \mathbb{R}^3$ as the rotary axis and $\alpha \in \mathbb{R}$ as the rotation angle, based on the Rodrigues' rotation formula [48], there exists infinite number of solutions to achieve identical result [49]. However, it should be noted that, all possible solutions of $\mathbf{a}$ stays within a 3-D plane π_n whose normal vector is exactly $\mathbf{n} = \pm({}^1\mathbf{e} - {}^2\mathbf{e})/||{}^1\mathbf{e}||||{}^2\mathbf{e}||$ [49]. If there are n_R vectors whose representations in $\mathcal{F}_1$ and $\mathcal{F}_2$ are available as

$$\begin{aligned} \mathbf{B} &:= \begin{bmatrix}{}^1\mathbf{e}_1 & {}^1\mathbf{e}_2 & {}^1\mathbf{e}_3 & \dots & {}^1\mathbf{e}_n\end{bmatrix}^\intercal \\ \mathbf{C} &:= \begin{bmatrix}{}^2\mathbf{e}_1 & {}^2\mathbf{e}_2 & {}^2\mathbf{e}_3 & \dots & {}^2\mathbf{e}_n\end{bmatrix}^\intercal \end{aligned} \tag{28}$$

where $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{n_R \times 3}$ are the corresponding two vector groups, we can define the cumulative error function based on the dot product of two vectors as follows:

$$E = \sum_{i=1}^{n_R} ||\mathbf{a} \cdot \mathbf{n}||^2 \tag{29}$$

where $\mathbf{n}$ denotes the (unit) normal vector of π_n. Differentiating (29) in terms of $\mathbf{a}$ and reformulate the result leads to

$$\underbrace{\begin{bmatrix} \sum_{i=1}^{n_R} x_i^2 & \sum_{i=1}^{n_R} x_i y_i & \sum_{i=1}^{n_R} x_i z_i \\ \sum_{i=1}^{n_R} x_i y_i & \sum_{i=1}^{n_R} y_i^2 & \sum_{i=1}^{n_R} y_i z_i \\ \sum_{i=1}^{n_R} x_i z_i & \sum_{i=1}^{n_R} y_i z_i & \sum_{i=1}^{n_R} z_i^2 \end{bmatrix}}_{\mathbf{N}} \underbrace{\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}}_{\mathbf{a}} = \mathbf{0}_{3\times 1} \tag{30}$$

where $\mathbf{a} \in \mathbb{R}^3$ is within the nullspace of matrix $\mathbf{N}$. One could then apply SVD to compute the best form of $\mathbf{a}$ in a closed-form manner. If we further denote α_i as the included angle between the orthogonal projection of ${}^1\mathbf{e}_i$ and ${}^2\mathbf{e}_i$ on π_n, the angle α is derived as the arithmetic mean

$$\alpha = \frac{1}{n_R}\sum_{i=1}^{n_R} \alpha_i. \tag{31}$$

Up to now, the rotation matrix between $\mathcal{F}_1$ and $\mathcal{F}_2$ is solved as $\mathbf{R}(\mathbf{a}, \alpha)$. Aiming for robot–camera calibration problem, the two coordinate frames explicitly correspond to the robot base frame and the camera, respectively. The vectors from $\mathbf{B}$ and $\mathbf{C}$ are, thus, n pairs of ${}^1\mathbf{e}_i = \mathbf{n}_{\zeta,i}$ and ${}^2\mathbf{e}_i = {}^c\mathbf{n}_{\zeta,i}$, and $\mathbf{R}(\mathbf{a}, \alpha)$ is exactly the robot–camera rotation matrix. As we have demonstrated the IP-based method to acquire one pair of them in Section IV, we only need to vary $\mathbf{R}_{\mathbf{q}_0}$ by additionally applying a fundamental rotation matrix. The assigned rotation angle for collecting each vector pair could be selected that distributes throughout 2π based on the total number of data to be acquired. This could minimize the in-between collinearity of the resultant projected VRLs on the image without leading the end-effector to exceed preset workspace by ϕ.

Remark 1: The process in (28)–(31) transfers the computation of the robot–camera rotation matrix by using n pairs of locally described vectors interpreted from IP-based feedback. The $\mathbf{n}_\zeta$ and ${}^c\mathbf{n}_\zeta$ are available via n times of independent IP-relevant control process, the solver avoids dealing with a complicated nonlinear problem. Its viability will be detailed in the following section.

Based on the computed $\mathbf{R}_c$, the translational component $\mathbf{t}_c$ of $\mathbf{T}_c$ should be further recovered. As a 3-D world point does not generate pose information between the robot and the camera, the estimation relies on the projected 2-D feature points. By initiating the robot end-effector to reach randomly selected goal states (whose leading $\mathbf{r}$ from $\mathbf{r}_0$ should also yield motion space constraint ϕ), a set of 2-D image points $\mathbf{y}_{t_1}$ and their corresponding configuration space data $\mathbf{x}$ are obtainable. Then, incorporating the calibrated $\mathbf{R}$ into the robot–camera transformation,

one could solve the remaining components for the calibration procedure by considering the following nonlinear optimization solver:

$$\min_{\mathbf{t}_c,\mathbf{t}_m} \quad \sum_{i=1}^{n_t} c_i \left(\mathbf{y}(\mathbf{q}_i,\mathbf{t}_c,\mathbf{t}_m)-\mathbf{y}_i\right)^2$$
$$\text{s.t.} \quad \mathbf{K}\mathbf{T}_c^{-1}(\mathbf{t}_c)(\mathbf{r}_i+\mathbf{t}_m)-f\leq 0 \tag{32}$$

where n_t is the preset number of data pair with $n_t \in \mathbb{N}^+, n_t > 2$, and c_i is the coefficient. The (nonlinear) inequality constraint appeared interprets a naturally enforced condition that the feature point must locate in front of the imaging sensor. The optimal $\mathbf{t}_c$ and $\mathbf{t}_m$ are solved upon (32) by computing Hessian matrix via dense quasi-Newton approximation [50]. Note that thanks to the computed $\mathbf{R}$, the minimization of 2-D feature differences only involves 3-D translations and could significantly reduce chances to reach local minima when recovering $\mathbf{t}_c$ and $\mathbf{t}_m$, compared to regarding $\mathbf{R}_c$ as the argument for estimation. Note also that the data collection in this part should reject cases where the 2-D feature points are collinear to avoid degenerated feedback input. To solve all elements in $\mathbf{t}_c$ of $\mathbf{T}_c$, the theoretical minimum number of data pair $\mathbf{y}_t$ and $\mathbf{x}$ should be 3.

Although the solver (32) finishes solving all components required for robot–camera calibration procedure, the $\mathbf{R}_c$ and $\mathbf{t}_c$ are solved separately, or notated as the $\mathbf{R}-\mathbf{t}$ approach. This is well known to potentially cause error propagation, which might result in low calibration accuracy especially for the end-effector position $\mathbf{r}$. However, as we will show that the abovementioned separated estimation approach could already provide a fairly accurate result without any initial guess, one can additionally apply an optimization step for combined regulation of $\mathbf{R}$ and $\mathbf{t}$ together based on the prerecorded data. The cost function writes $\min_{\mathbf{R},\mathbf{t},\mathbf{t}_m} c\sum_{i=1}^{n_t}\left(\mathbf{y}(\mathbf{q}_i,\mathbf{R},\mathbf{t},\mathbf{t}_m)-\mathbf{y}_i\right)^2$ and is deployed by considering the previous calibration result as initialization. We notate this improved version as $\mathbf{R}-\mathbf{t}-\mathbf{T}$ approach. Such refinement of estimation could improve the estimation robustness to measuring noises and kinematics model error of the robot, which are unknown and highly coupled in our model. The performance before/after combined optimization will be shown in the following section.

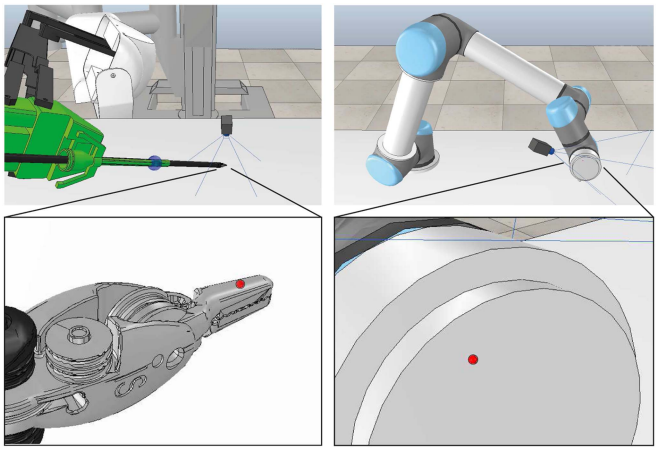
C. Solvability

In order to provide valid robot–camera calibration results, the solvability of the abovementioned scheme should be particularly investigated. In terms of the rotational components in $\mathbf{R}$, by using the IFP modeling, the matrices $\mathbf{B}$ and $\mathbf{C}$ must contain, at least, two noncollinear vectors, or $n_R \geq 2$. In addition, at least one pair of vectors should not be collinear to each other. Thus, the rank of $\mathbf{B}$ and $\mathbf{C}$ constructed by data input should be not less than two as well. This indicates at least two independent control processes subject to $\mathbf{u}$ should be deployed to minimize d while assigning different $\mathbf{w}_\zeta$. If $n_R > 2$, the solver for (30) using SVD provides data fitting as of least-square method, which is obviously solvable. It should be noted that each $\mathbf{n}_{\zeta,i}$ and ${}^c\mathbf{n}_{\zeta,i}$ might exist two possible solutions as the normal vectors of the converged IFP. However, this could be simply sifted by computing the dot product to enforce acute included angles (invert the vector if meeting obtuse angle). Meanwhile, the estimation of the translation part via optimization in (32) uses the computed $\mathbf{R}_c$ as a constant input. Compared to the direct blind estimation of all components in $\mathbf{T}_c$ without a reasonable initialization, a computed $\mathbf{R}_c$ facilitates more efficient iterations and less likelihood to be trapped in local minima, as the 2-D reprojection errors of the feature point along different robot configurations is now only attributed to 3-D translation differences. The local minima of the cost function could, thus, be significantly reduced to avoid contradictory iterative directions of the parameters while minimizing individual feature point errors.

In terms of the $\mathbf{R}-\mathbf{t}-\mathbf{T}$ approach, the final phase of estimation refinement regards both $\mathbf{R}$ and $\mathbf{t}$ as the tuning arguments to further minimize the residual image-based errors. As the elements in both $\mathbf{t}$ and $\mathbf{t}_m$ should be recovered, the number of independent parameters to be solved is six, which indicates a minimal three groups of data (robot configuration $\mathbf{q}_i$ and feature projection $\mathbf{y}_i$) suffices the estimation. At this stage, the previous calibration results from the $\mathbf{R}-\mathbf{t}$ method is used for initializing the nonlinear optimization as well. This allows the iterations to operate within local domain of the parameters around their true values. Thus, the overall pipeline for solving robot–camera calibration could be deployed as long as the collected data meets the abovementioned needs.

VI. Simulation Results

In this section, we first conduct simulation study to quantitatively investigate the performance of our algorithm under different parameter set-ups. We use the CoppeliaSim v4.4.0 (Coppelia Robotics, Ltd.) robotics simulator as the virtual validation environment, which interfaces to MATLAB R2020b (MathWorks Inc) via external application programming interface. To comprehensively validate our approach, without loss of generality, we adopt the models of two popular robotic systems to the virtual environment. One is the da Vinci research kit (dVRK) [51] (with its simulation model further referring to [52]) developed from the dVSS (Intuitive Surgical Inc.), which is currently a paradigm surgical robotic system worldwide. The other is the UR5 by the Universal Robots, which has the typical kinematic design of 6-DoF serial revolute joints. The feature point for providing visual feedback is fixated at an arbitrary (distal) position attached to the robot end-effector, such that it could be actuated by all robot joints. The feature point, subject to the robot's initial configuration, should be observable by a monocular camera with known intrinsic parameter matrix $\mathbf{K} \in \mathbb{R}^{3\times4}$ as an eye-to-hand set-up. The resolution of the camera is set to 640×480 pixels. We again emphasize that the 3-D position of the feature point $\mathbf{t}_m$ does not need to be known in all set-ups. The feature is a red sphere (see Fig. 4 for illustration) with diameter of 1 mm, which is detected by computing the centroid of the region on the image subject to color-based segmentation. This implies that despite simulation, there exists a theoretical sensing error in the robot–camera model, as the 2-D centroid is not necessarily the exact projection of the center of the 3-D sphere. The detailed



(a) (b)

Fig. 4. Two simulation set-up scenes using two typical types of robot manipulators and the cameras (in gray boxes). The red points in the magnified view are the observed world feature point fixed on the end-effectors, respectively. (a) Simulation set-up with dVRK. (b) Simulation set-up with UR-5.

TABLE I
PARAMETER SET-UPS OF THE SIMULATION SCENE FOR ROBOT–CAMERA CALIBRATION USING DVRK AND UR5, RESPECTIVELY (THE UNIT FOR TRANSLATIONAL COMPONENTS ARE ALL MM

Scene	dVRK	UR5
$\mathbf{R}$	$\begin{bmatrix} -0.927 & 0.303 & -0.220 \\ 0.372 & 0.806 & -0.460 \\ 0.038 & -0.508 & -0.860 \end{bmatrix}$	$\begin{bmatrix} -0.886 & -0.306 & 0.349 \\ -0.457 & 0.446 & -0.769 \\ 0.080 & -0.841 & -0.535 \end{bmatrix}$
$\mathbf{t}$	$\begin{bmatrix} -100.309 & 6.265 & 29.919 \end{bmatrix}^\intercal$	$\begin{bmatrix} -40.368 & -40.897 & 0.298 \end{bmatrix}^\intercal$
$\mathbf{t}_m$	$\begin{bmatrix} 1.569 & 6.343 & 0.610 \end{bmatrix}^\intercal$	$\begin{bmatrix} -18.766 & -13.052 & 20.814 \end{bmatrix}^\intercal$
${}^c\mathbf{m}_0$	$\begin{bmatrix} 1.541 & -11.536 & 119.30 \end{bmatrix}^\intercal$	$\begin{bmatrix} 2.203 & -4.605 & 10.047 \end{bmatrix}^\intercal$

set-up parameters for the robots could be referred to Table I, which are trivial with nonparticular as the general situations, but is reasonable for application scenarios.

The VFL could be arbitrary selected in 3-D space as long as reachable by the robot's motion space. Throughout the simulation and experiments, for simplicity, we set the VFL to be parallel to the x-axis of the robot end-effector to avoid trivial intermediate transformation calculus. Then, we could adopt the following trajectory:

$$\bar{\mathbf{r}}_\zeta = \begin{bmatrix} \cos(s(t)) & \sin(s(t)) & 0 \end{bmatrix}^\intercal \tag{33}$$

whose dynamics is further parameterized by the corresponding $s(t)$ with respect to $\mathbf{r}_0$ as

$$s = \pi \frac{1 - \cos\varpi t}{2}. \tag{34}$$

The leading geometry of the path ζ by (33) is, thus, a half-circle curve, which is obviously continuous and convex that satisfies the properties defined in Section IV-A. Meanwhile, $s(t) \in [0, 1]$ in (34) also allows periodical motion of the robot end-effector along the path to generate IP movements.

We first validate how different amount of data (by tuning n_R and n_t) will affect the calibration accuracy under an identical set-up. The combined approach, or the $\mathbf{R} - \mathbf{t} - \mathbf{T}$ pipeline is used at this stage. We assume the available motion space for the robot end-effector is a sphere with radius of only 10 mm (i.e., $\phi = 10$ mm), which is significantly smaller than the available motion space of the robot itself. Then, by deploying the abovementioned parameter set-up, we assign a random and different $\mathbf{w}_0$ for each IFP convergence step and complete 50 individual robot–camera calibration process. An average L^2-norm rotation/translation calibration error (rotation error computed using Euler angles) is then computed. Fig. 5 demonstrates the performance of the single-point robot–camera calibration accuracy under restricted robot motion space of 10 mm but with different number of data n_R and n_t. It is easy to discover that, generally, increasing the number of either data not only decreases the average calibration errors but also reduce the result uncertainties (inspected by the span of error bars). Particularly, by selecting $n_R \geq 8$ and $n_t \geq 8$, the average accuracy of both rotational and translational components have further decreased, i.e., $< 0.3^\circ$ in rotation and < 0.7 mm in translation regardless of the robot being tested. When n_t is set to 20 in the UR5 case, the calibration error decreases to $\sim 0.09^\circ$ for 3-D rotation and ~ 0.11 mm for 3-D translation even n_R is only set to 20. Note that the centroid-based 2-D feature detection introduces theoretical sensing error, which is commonly encounter in realistic application. Meanwhile, it is notable that, the calibration accuracy is weakly reflected by the corresponding reprojection error of the feature point on the image, especially when n_R and n_t are smaller than 8. The basic average error for different number of data around 0.11 px for the dVRK case and 0.07 px for the UR5 case.

We then conduct comparative analysis of our approach to study the calibration accuracy under different properties. The number of collected data is now set to $n_t = 20$ with varying n_R. Again, we do 50 trials for each identical set-up. First, we vary ϕ among 5 mm and 20 mm to investigate the influence of the restricted motion space on the calibration accuracy. Fig. 6 shows the calibration result relationship among different ϕ and n_R. We can see that larger motion space tends to result in lower accuracy under the identical n_R, which is reasonable as it performs more informative spatial data input for calibration. As $\mathbf{W}_0$ and the data for estimating $\mathbf{t}$ are both randomly selected, the curves slightly fluctuate but still show clear trends. For $\phi \geq 10$ mm, the average rotation error tends to be around 0.07° with n_R being greater than 10.

Next, we fix the value of ϕ and apply different sensing noises of the feature point on the image. Such investigation is necessary as image-based measurements inevitably include noises, which might affect the feedback quality, especially for a single point feature available in our case. We generate a random noise on the observed position of the 2-D feature point from the image with its distribution with respect to 0 being $U(-\beta, \beta)$, which is a uniform distribution. Fig. 7 shows the results of the calibration accuracy under noises with different magnitudes. The larger the noise magnitude is set, the larger the rotation and translation error exhibits. However, under each case, the accuracy appears to be reliable across different values of n_R being selected. The

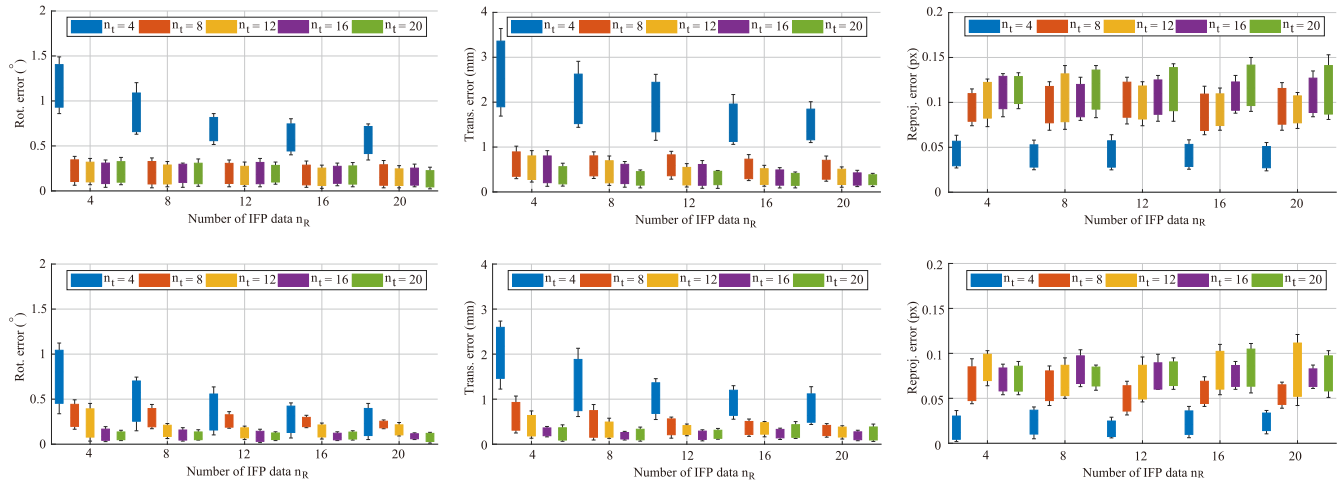


Fig. 5. Demonstration of robot–camera calibration performance under different amount of data using n_R and n_t. The data in the first/second row indicate the validation using the virtual dVRK/UR5 as the robot, respectively. In each row, the average of the norm of rotation error, translation error, and feature back-projection error is shown in the left, middle, and right subfigure, respectively.

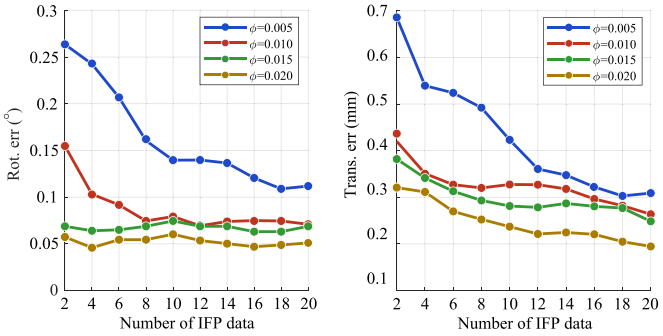


Fig. 6. Performance analysis of setting different motion space restrictions via ϕ and the corresponding calibration errors (left: rotation error, right: translation error) on dVRK. For each set-up, we did five trials and compute the average errors.

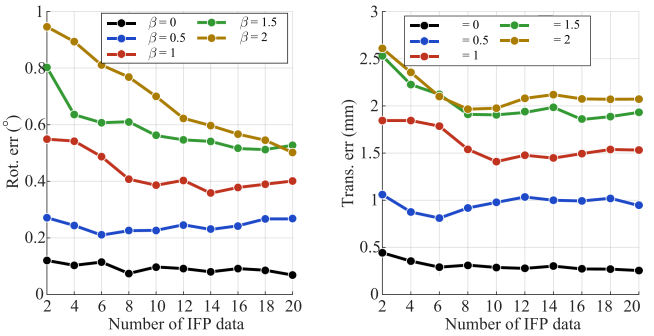


Fig. 7. Performance analysis of setting different magnitudes of random noises and the corresponding calibration errors (left: rotation error, right: translation error) on dVRK. For each set-up, we did five trials and compute the average errors.

most stable behavior of the residual error occurs at the case with no noise. This indicates that the noises impose explicit affect on the calibration accuracy, and the fluctuation is also positively correlated as well.

Now, we further conduct comparative analysis to show the calibration performance using different parameter estimation pipelines. As stated, we could recover the robot–camera transformation using $\mathbf{R}-\mathbf{t}$ approach the $\mathbf{R}-\mathbf{T}$ approach. We first apply the same motion space restrictions and then record the calibration results via different n_R. Fig. 8 shows the results of the calibration errors using box charts. Generally, larger motion space leads to better calibration accuracy for both pipelines, which is consistent to the previous results. However, the $\mathbf{R}-\mathbf{T}$ approach outperforms the $\mathbf{R}-\mathbf{t}$ approach in terms of both the rotation and translation errors, and own smaller uncertainties under different n_R. While using the $\mathbf{R}-\mathbf{t}$ approach, as the rotation error directly propagates to the translation component estimation phase, increasing the number of IFP data leads to a larger and fluctuated estimation errors. The $\mathbf{R}-\mathbf{t}$ pipeline otherwise provides much more accurate and consistent performance in eliminating residual feedback errors.

Then, we apply random noises generated toward image features with different magnitudes, with the results being shown in Fig. 9. The performance is similar to that of changing ϕ, where the combined approach exhibits smaller and more consistent calibration errors compared to the separated pipeline. However, the increasing magnitude of the noise might also render the $\mathbf{R}-\mathbf{T}$ pipeline fluctuate more, but still performs better than the $\mathbf{R}-\mathbf{t}$ pipeline. This proves our calibration method, despite use of a single feature point, an average 0.45° rotation error and 1.51 mm translation error could be reached upon the image sensing noise of ± 1 pixel, with good consistency as well.

VII. Experiment Results

A. Overview

We tested our robot–camera calibration algorithm using the dVRK because robotic surgery is typically performed in a tightly-constrained environment that lacks consistent visual features. We will first show the basic performance of our algorithm that could stably acquire visual data by individual IP-based

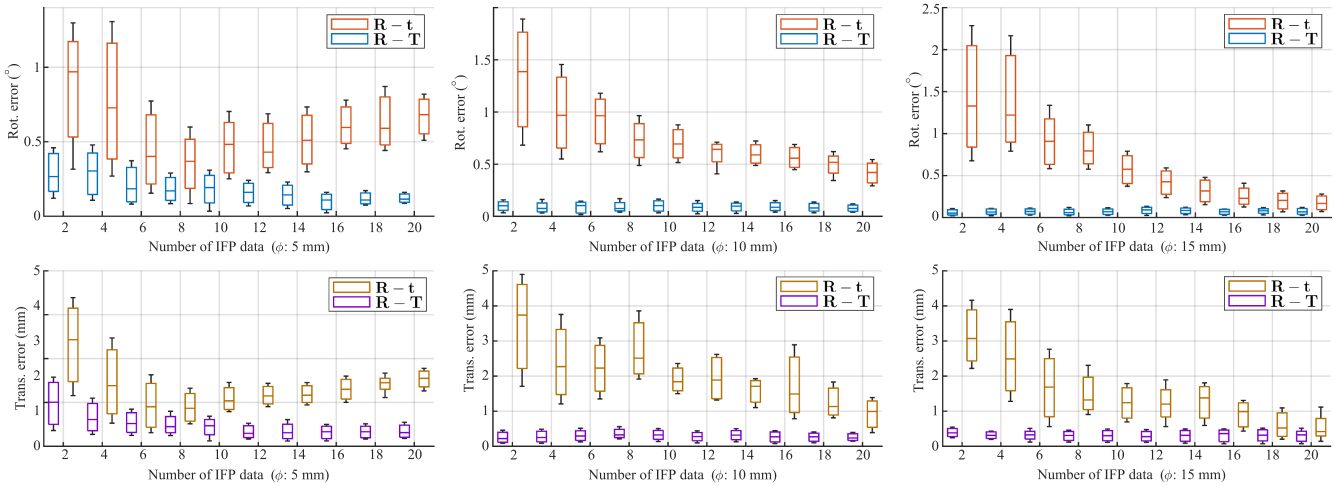


Fig. 8. Comparison of calibration accuracy on dRVK under different motion space restrictions ϕ (first/second row: position/orientation error. SEP and SEP+COMB indicates calibration results using $\mathbf{R} - \mathbf{t}$ and $\mathbf{R} - \mathbf{T}$ pipeline, respectively.

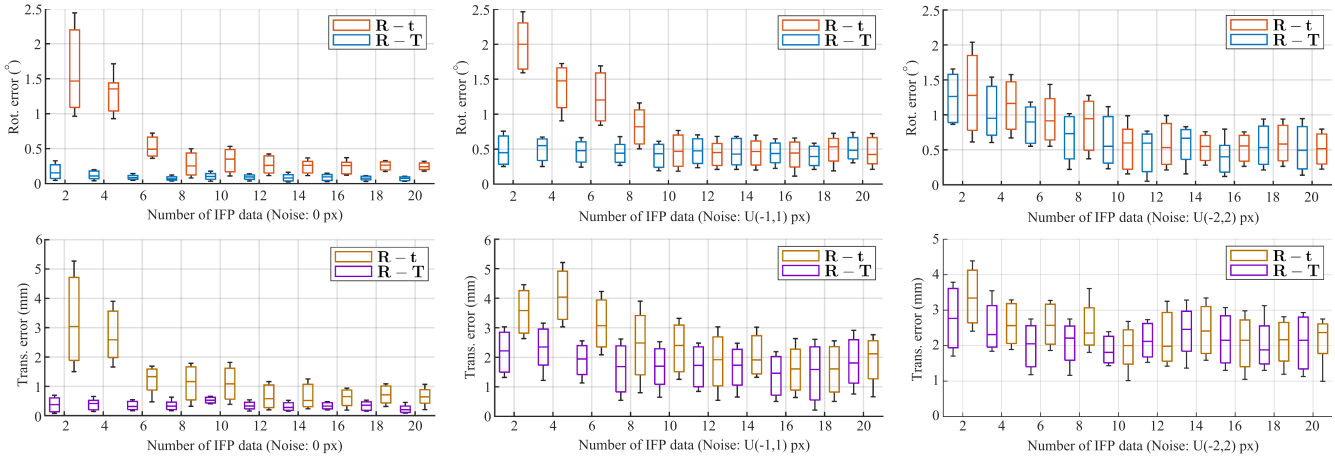


Fig. 9. Comparison of calibration accuracy on dVRK under different magnitudes of uniformly distributed random noises (first/second row: position/orientation error). SEP and SEP+COMB indicates calibration results using $\mathbf{R} - \mathbf{t}$ and $\mathbf{R} - \mathbf{T}$ pipeline, respectively.

control progresses subject to restricted motion space. Then, we conduct comparative analysis with existing popular methods, and also with our approach under different computation pipelines. The metrics used for performance evaluation include back-projection of robot end-effector on 2-D image upon collected kinematics data, and also 3-D measurement of robot tip positioning accuracy over prescribed measurable world positions.

B. Set-Up

The dVRK is formed by a set of 6-DoF active serial joints for actuating a robotic surgical instrument, with an additional DoF to control the jaw opening angle of the distal tool. Throughout our experiments, we use the large needle driver as the mounted instrument type and keeps the jaw angle of the driver to 0, such that it is equivalent to a 6-DoF general robot arm. To recall the surgical set-up, the distal tip is usually extended from the RCM point for around 100 mm [53] as a common distance to reach the (surgical) operation space. Due to the tendon-driven design of the instrument, the end-effector positioning accuracy might be intrinsically lower than that of the industrial robot arm [54]. However, we clarify that all the model errors will not be compensated in advance, which are to be reflected by the visual features and will be collected by our algorithms. Both the integrated laparoscopic camera and an industrial camera will be used as our imaging sensors in different set-ups. The resolution for both the industrial camera and the laparoscopic camera are set to 640 × 480 pixels. Fig. 10 shows the layout of our robot–camera set-up.

We only use one single world feature point as visual feedback during the experiment. It is manually selected by the user using mouse clicking on the screen as an initialization step. This feature, now described in 2-D pixel (as $\mathbf{y}$), is to be tracked during the data collection step using an optical flow algorithm, which is a mature image tracking method. We again emphasize that, the 3-D information of the feature point remains unknown (as we are using sole monocular images), which is also not required

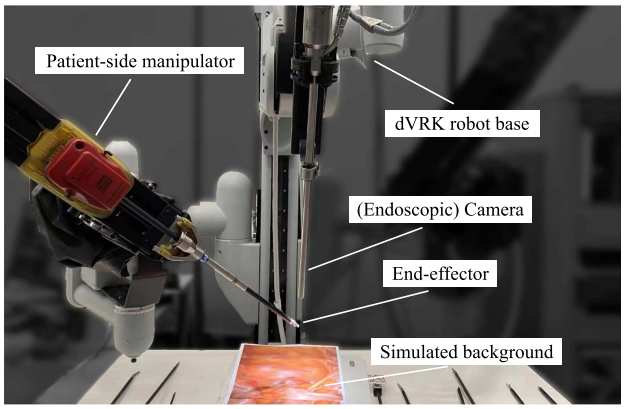


Fig. 10. Experimental set-up using dVRK as the robot platform.

TABLE II
PARAMETER SET-UPS FOR DATA ACQUISITION IN EXPERIMENTS

Parameters	Values
$\mathbf{r}_0$	$\begin{bmatrix}-131.92 & -16.34 & -106.08\end{bmatrix}^\intercal$ mm
$\mathbf{w}_0$	$\begin{bmatrix}-1.642 & -0.243 & 2.268\end{bmatrix}^\intercal$ rad
ϕ	10 mm
$\mathbf{\Gamma}$	$\text{diag}(0.05, \ldots, 0.05) \in \mathbb{R}^{6\times 6}$
γ	0.01
$\overline{\omega}$	0.03
κ	0.1
t_{m_0}	$\begin{bmatrix}0.01 & 0 & 0\end{bmatrix}^\intercal$
t_0	$\begin{bmatrix}0.01 & 0 & 0\end{bmatrix}^\intercal$
c_i	$0.1\ \forall\ i \in [i,\ n_t]$

throughout the calibration procedure. However, note that the 3-D position of the feature point will finally be automatically identified together with the robot–camera transformation matrix $\mathbf{T}$ after the calibration is completed.

We use a PC with Core i7 3.4 GHz desktop CPU and 128 GB RAM as the upper level controller. The computer connects to the robot controller via TCP/TP, and the software communication is completed by the cisst/SAW libraries and dVRK ROS MATLAB, which is run on MATLAB R2020b (MathWorks). Note that we do not deploy any acceleration schemes (either parallel computation or GPU array) as our algorithm is not computationally expensive. The camera is precalibrated with known distortion parameters and captures online streaming videos at ∼30 fps, while the lens distortion has been rectified to improve the sensing accuracy.

C. Data Acquisition Performance

We first evaluate the performance for our IP-based data acquisition scheme and its leading robot–camera calibration performance upon typical set-up. We assign the motion space restriction of the robot end-effector as $\phi = 10$ mm, which indicates a 10 mm radius sphere. This is a significantly restricted scenario compared to that required by existing calibration approaches where calibration object itself is already at such scale [55]. We assign $n_R = 5$ and $n_t = 20$, as from the simulation part, such amount of data is reasonable to achieve both efficient data acquisition process and decent calibration results under model uncertainty. After convergence during each process, the robot additionally applies a rotation matrix in euler form as $[0\ \ 0\ \ o_p]^\intercal$ from the last $\mathbf{w}_0$ to deviate the 3-D position of m upon $s = s_s$ and s_t to avoid generating repetitive data input. o_p is further computed as $o_p = \pi/n_R$ to maximize the visual differences subject to restricted ϕ. The assignment of other parameters are shown in Table II. When collecting the n_t number of data for computing $\mathbf{t}$, we assign the end-effector position to uniformly and randomly distributed in the sphere to maximize the use of the motion space. The initial state of the robot end-effector $\mathbf{r}_0$ and $\mathbf{w}_0$ again yields a typical set-up in robot-assisted surgery. The robot–camera relationship allows the camera to observe the robot's distal tip around the center of the image, such that it tends to stays within the field of view of the camera.

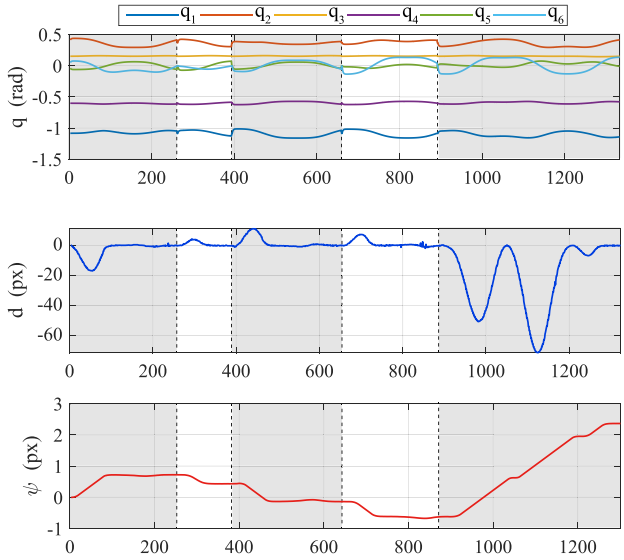


Fig. 11. Evolution of robot joint positions $\mathbf{q}$, IFP orientation ψ, the image-based feature-to-reference distance d upon IP-based IFP regulation with $n_R = 5$, $\phi = 10$ mm, which indicates five individual convergent processes (Shaded areas represents first, third, and fifth convergent process).

Fig. 11 illustrates the robot motions subject to IP during data acquisition for $\mathbf{B}$ and $\mathbf{C}$. The robot has completed totally five independent control processes with different n_p, which are separated with shaded and unshaded areas along time steps. Under a 30 Hz overall control loop frequency, in each control process, the image-based point-to-line error d in (10) is stably minimized to 0 subject to the IP-induced motion. After convergence of d, the robot end-effector motion continues due to the periodical movement of the feature point along ζ upon dynamics of s. Meanwhile, ψ is also stably converged each time upon regulation of d, and is smoothly changed throughout the evolution, as for simplicity, we directly use the last settled ψ to initialize the next control phase. Once the variation of ψ is small enough (judged by a preset threshold), ${}^1\mathbf{e}$ and ${}^2\mathbf{e}$ are computed and stored, and then a new control process starts along with a newly set o_p. Note that, the stable convergence of d is guaranteed by the definition

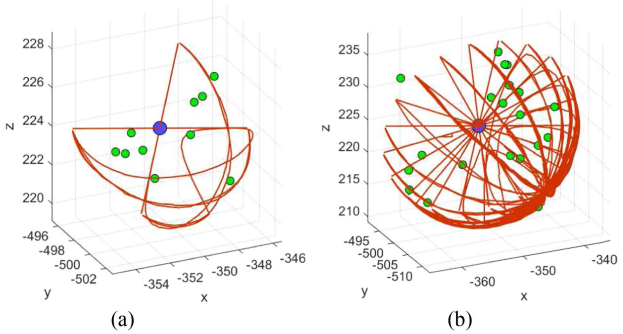


Fig. 12. Recorded robot end-effector motions during data acquisition under different software set-ups. Red lines: robot end-effector trajectories; green points: data samples for estimating $\mathbf{t}$. (a) $\phi = 5$ mm for motion space restriction, $n_R = 2$ and $n_t = 10$ for the number of data samples. (b) $\phi = 15$ mm for motion space restriction, $n_R = 10$ and $n_t = 20$ for the number of data samples.

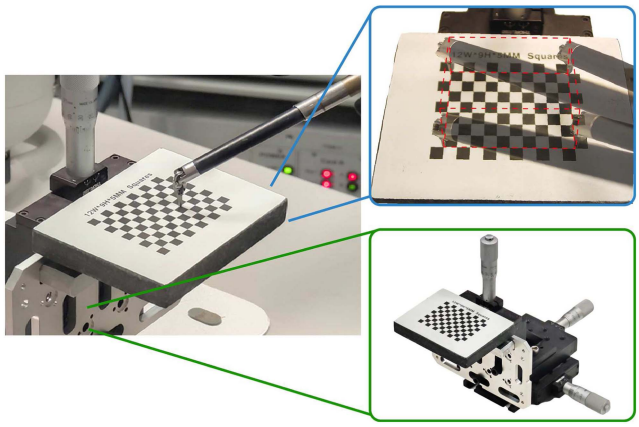


Fig. 13. End-effector measurement set-up (left) using a checkerboard mounted on an XYZ microtrimming platform (bottom right) to generate measurable world position reference points using corner pixels within the volume enclosed by the red dash lines (bottom right), which are to be reached by the robotic end-effector tip.

of a convex trajectory used for IP-based motions, as this rejects the 2-D position of $\mathbf{y}$ on the image to go across l_r under a fixed ψ. This is important to allow that the image-based regulation of ψ for minimizing d owns a stable equilibrium point. The evolution of d shows small fluctuations, which are attributed to sensing noise on the image while tracking $\mathbf{m}$ but stably converges and indicates good robustness of our controller. The time cost to reach convergence in each process is different, which is affected by the differences of initial d.

We also show the robot end-effector's motions under SE(3) space throughout the data acquisition process in Fig. 12. The two cases show that regardless of the number of data assigned by n_R and n_t, the envelopes of the trajectory stay within the motion space exactly defined by ϕ without applying additional constraints. In detail, the data acquisition phase of rotation (through the red trajectories in Fig. 12) are continuous and occupies the semisphere of the target workspace, while the discrete target positions for estimating the overall robot–camera transformation and randomly selected within the sphere.

D. Comparative Evaluation

We then evaluate the calibration accuracy and conduct comparative analysis with different approaches and our method with different pipelines. As we only use a single feature point as the sole visual feature, typical validation approaches, and datasets relying on calibration objects and/or landmarks for accuracy evaluation becomes not applicable for our evaluation. Meanwhile, the inaccurate robot kinematics model due to tendon-driven actuation makes the measurement of the ground truth unreliable if using external sensors [55]. Thus, for accuracy validation, we aim to measure the 3-D positioning accuracy of the end-effector, affected by both $\mathbf{R}_c$ and $\mathbf{t}_c$, by comparing the error with predefined measurable ones. We construct a calibration plate with a grid to be reached by the end-effector as shown in Fig. 13 . The plate is further mounted on an XYZ microtrimming platform with two functions: 1) to adjust the plate vertically to let sample points cover a volume instead of a planar surface to generate more error data, and 2) to allow precise measurement of the position error between the end-effector and the target corner pixel. The plate grid is of 12×9 with 5 mm for each square. As the movable range of the platform along z-axis is 30 mm, one can generate overall 352 calibration reference points (using world corner pixels) within a volume of $55 \times 40 \times 20$ mm if we use 20 mm with 5 mm interval for z-axis tuning. Their 3-D positions are all measurable from the camera. After the end-effector is directed to that position using calibrated $\mathbf{T}_c$, its positioning error to the reference position is measured by manually trimming the plate position to eliminate the residual distance between the end-effector and the corresponding corner pixel. To meet practicality, we assign $n_R = 5$, $n_t = 20$, and $\phi = 10$ mm as the software set-up.

We validate the calibration accuracy from two aspects: the 3-D positioning error of the end-effector, and the 2-D back-projection error of the distal tool point on the image. We compare the result with that of one of the most commonly used robot–camera calibration approaches, i.e., the Tsai's method [15]. To do so, we additionally use an 8×5 checkerboard grid with 4-mm square size mounted on the instrument and capture 20 images for calibration. We totally compare the following four methods:

1) the standard Tsai's method using the checkerboard using 20 images;
2) the standard Tsai's method as initialization, followed by nonlinear optimization using back-projection error from data acquisition via random positioning, as optimization-based estimation is commonly used in markerless situations;
3) our method with $\mathbf{R} - \mathbf{t}$ pipeline;
4) our method with $\mathbf{R} - \mathbf{T}$.

Fig. 14 visualizes the envelope of the tight motion space ($\phi = 10$ mm) complied by the robot end-effector during data acquisition in different cases. The back-projected sample points for refining $\mathbf{T}_c$ are also visualized. Fig. 15 shows the measured 3-D positioning errors of the end-effector using the microtrimming platform. The error using $\mathbf{R} - \mathbf{T}$ pipeline is generally smaller than that of $\mathbf{R} - \mathbf{t}$ and Tsai's$-\mathbf{T}$. The results from $\mathbf{R} - \mathbf{t}$ spread wider especially along z-axis of the grid. Fig. 16

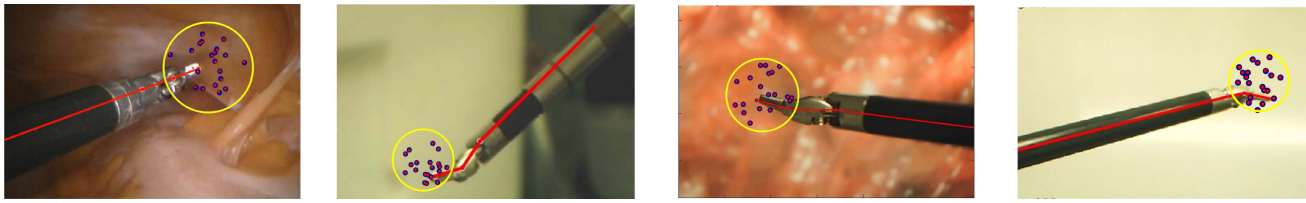


Fig. 14. Back-projection of the random sample points (red points) in all four set-up cases using our calibration method (s.t. $n_t = 20$) and the collected 2-D image feedback (blue points), the yellow circles show the projected restricted motion spaces during data acquisition phase on the image, which are all a 10-mm-radius sphere (s.t. $\phi = 10$ mm) in our setting.

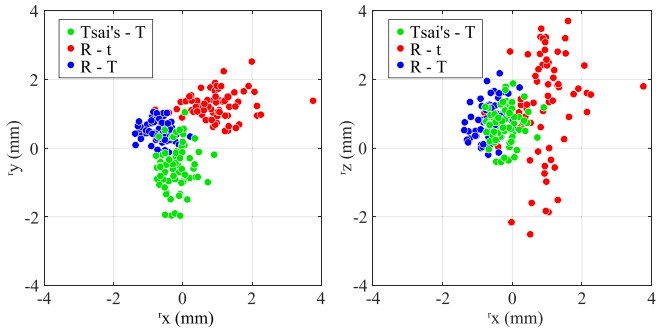


Fig. 15. Superpositioned measured positioning errors of the end-effector tip with respect to the corresponding reference points using three methods, visualized with respective to the checkerboard frame toward X–Y and X–Z plane, respectively.

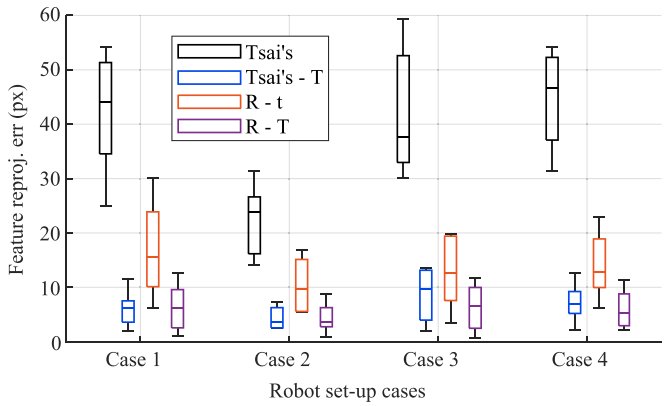


Fig. 17. 2-D error on image.

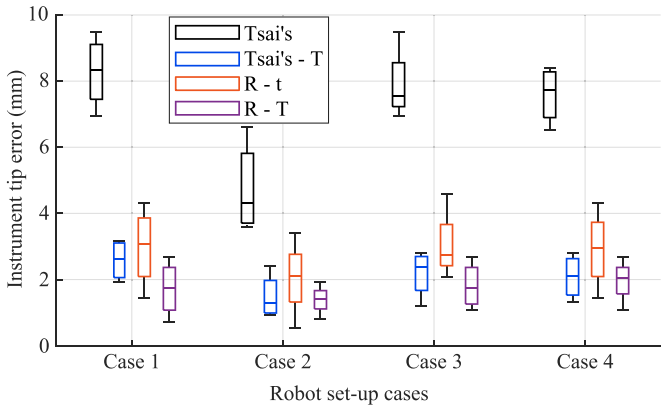


Fig. 16. 3-D error on image.

further shows the norm of the 3-D errors using four different pipelines. The results using the standard Tsai's method in all four cases have the largest back-projection errors and poor consistency, which is likely to be caused by the enlarged robot kinematics error led by the mounting calibration grid, or the small motion space restricted by the camera's field of view ($\sim 40°$). However, by applying an additional data acquisition and optimization step using the standard Tsai's method as an initial guess, the errors are significantly reduced from around 6.9 ± 3.3 mm down to 2.2 ± 1.3 mm from four cases. Meanwhile, our method using $\mathbf{R} - \mathbf{t}$ pipeline performs better than Tsai's method, and $\mathbf{R} - \mathbf{T}$ outperforms $\mathbf{R} - \mathbf{t}$ in both the accuracy and consistency. This is because the optimization of $\mathbf{T}_c$ avoids the rotation error to be propagated to the end-effector position. The results show that the $\mathbf{R} - \mathbf{T}$ pipeline of our method using a single feature point is competitive to the Tsai's + $\mathbf{T}$ method using calibration objects with more accurate and stable robot positioning. The average 3-D positioning error upon calibration using $\mathbf{R} - \mathbf{T}$ pipeline across four cases is 1.8 ± 1.0 mm, compared to the 2.9 ± 1.9 mm by $\mathbf{R} - \mathbf{t}$ pipeline, and 2.3 ± 1.2 for the Tsai's + $\mathbf{T}$ method. Fig. 17 shows the back-projection results using the four method applied to four different set-up cases using the reference positions. The performance comparison is similar to that of the 3-D positioning errors. Our $\mathbf{R} - \mathbf{T}$ pipeline performs best in all four pipelines with an average 6.5 ± 7.8 px back-projection errors, while the result of $\mathbf{R} - \mathbf{t}$ pipeline is 13.8 ± 15.8 px. Note important that, the result from $\mathbf{R} - \mathbf{T}$ does not rely on any initialization of $\mathbf{T}$. Fig. 18 finally demonstrates the visualized back-projection results of the robot skeleton using the robot–camera calibration results using our method.

We also compare the computation performance using the abovementioned methods by evaluating their abilities to provide converged results to reference robot–camera transformation. By maintaining the previous system set-up and the amount of data collection, we do 50 trials for each case and set the outlier threshold as the L^2-norm rotation or translation error relative to the reference $\mathbf{T}_c$ to be greater than 100 mm/1 rad, respectively. As ground truth is not available, the reference $\mathbf{T}_c$ is computed by averaging the inliers. We then calculate the percentage of the outliers as the convergence rate, where the results are shown

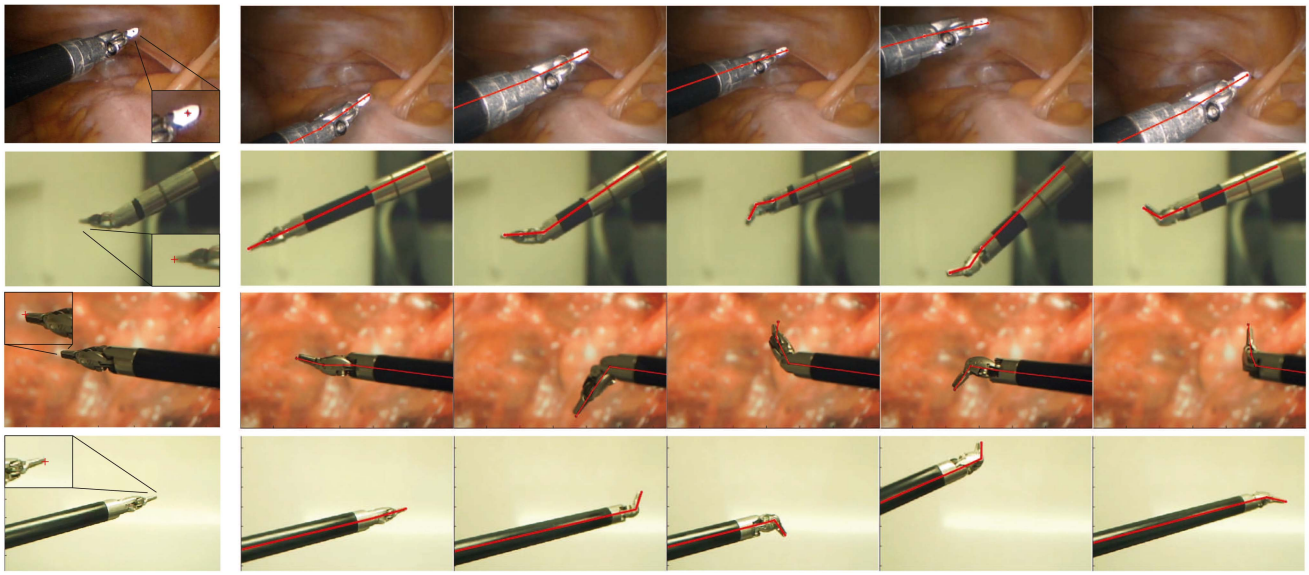


Fig. 18. Illustration of the robot skeleton back-projection results overlaid on the raw sensing camera images with random configurations. The four rows of figures indicate the four different set-up cases. The first column shows the selected feature point (by green circles) in each case. The red lines and red dots show the back-projected robot links and skeleton nodes, respectively.

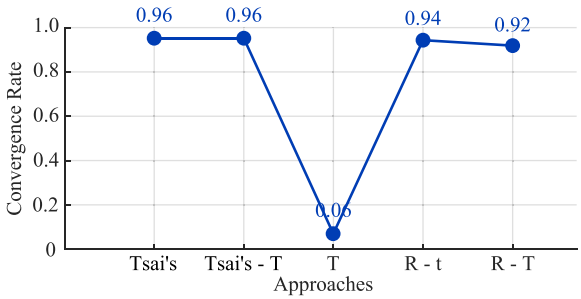


Fig. 19. Convergence percentage of different approaches while commencing 50 individual trials upon $n_R = 5$, $n_t = 20$, and $\phi = 10$ mm. For direct estimation of $\mathbf{T}$, we randomly initialize it as the lack of any knowledge.

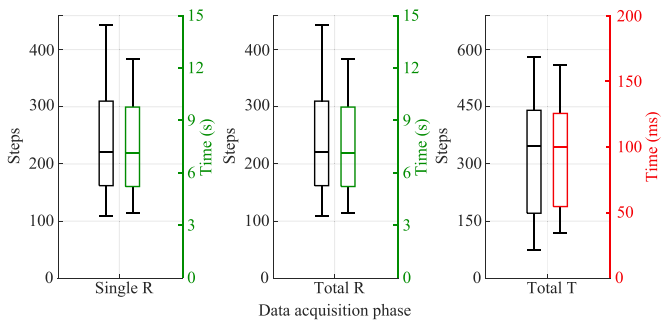


Fig. 20. Steps and time cost for: collecting one pair of ${}^1\mathbf{e}$ and ${}^2\mathbf{e}$ for estimating $\mathbf{R}_c$ (left), collecting the data matrix $\mathbf{B}$ and $\mathbf{C}$ (middle), and the optimization process for refining $\mathbf{T}_c$.

in Fig. 19. Our method using $\mathbf{R} - \mathbf{t}$ and $\mathbf{R} - \mathbf{T}$ pipeline own convergence rate of 47/50 and 46/50, respectively, which is as good as the Tsai's method being 48/50. Adding estimation step of $\mathbf{T}$ to Tsai's method does not affect the rate, while slightly decreasing to that using IP-based estimation. The convergence rate of our algorithm using a single feature point is similar to the Tsai's method. The possible explanation for them not reaching 100% could be the optimization process under random pose selection might occasionally be degenerated in the observable data space. We additionally apply optimization-based estimation of $\mathbf{T}_c$ from back-projection feature errors with blind initialization of $[1\ 0\ 0]^\intercal$ for all $\mathbf{R}_c$, $\mathbf{t}_c$, and $\mathbf{t}_m$, respectively. The estimation result shows 3/50 convergence rate, far worse than that by the others, owing to the unpredictable local minima appeared during iterations from poor initialization. It implies that our method owns good performance while avoiding local minima using the IP-based data collection scheme under limited number of data.

We additionally compute the time cost for different phases to show the consistency in efficiency. We fix the original set-up and apply different initial orientation of the IFP with a random but nonzero initial ψ. Fig. 20 demonstrates the consumed steps and equivalent time cost to finish the data collection and computation process. The mean value for a control process for collecting a pair of ${}^1\mathbf{e}$ and ${}^2\mathbf{e}$ is 213 ± 125 steps and 7.1 ± 5.4 s. As $n_R = 5$, the overall IP-based control processes is 1260 ± 597 steps and 35.8 ± 9.2 s, respectively. This indicates around 7 s required to converge one pair of vectors. The reason for the inconsistency across individual process is mainly due to the different convergent paths of ψ from different (randomly) initialized values. Note that the computation of $\mathbf{R}$ after acquiring the data matrix $\mathbf{B}$ and $\mathbf{C}$ using SVD is analytical whose computation time is negligible. The corresponding control frequency is ~ 29 Hz. The average durations for computing the final $\mathbf{T}_c$ is 87 ± 82 ms. We again emphasize that the above time cost are computed without software acceleration, whose actual consuming time could be improved, but is considered negligible compared to the IP control phase.

VIII. Discussions

A. Result Performance

We have quantitatively demonstrated the calibration performance of our method and compared it with two typical types of approaches: the standard approach that relies on calibration object as the Tsai's method, and the offline–online combined pipeline as Tsai's+$\mathbf{T}$ method. First, it is important to note that, our algorithm is executable within a 10-mm-spherical end-effector motion space, which shows distinct potential to adapt to tightly constrained environment. No studies have explicitly tackled such constraint for robot–camera calibration. However, it has been shown that our method outperforms both these two types with lower 3-D positioning errors and 2-D back-projection errors. It is also better than existing state-of-the-art approaches evaluated on the identical dVRK platform (e.g., 20 mm in [40], >10 mm in [39], 2.0 mm in [41], and 3.2 mm using our previous work using line features [12]). These approaches use the estimation error of $\mathbf{t}_c$, which should have indicated smaller errors compared to our metric (i.e., the 1.8 mm end-effector positioning error), as it introduces additional error propagated by $\mathbf{R}_c$. Our SVD-based solver leads to competitive convergence rate (reduced possibility to reach local minima) as the Tsai's method (i.e., 96%) despite no initialization. This generally shows that our method not only systematically reduces the constraints required but improving calibration results from different aspects as well, with good consistency under sensing noises and modeling error, even though the single-point feedback input is generally more vulnerable to disturbances.

Currently, there are no existing datasets established for robot–camera calibration applicable to be deployed by our method. An important reason is that, our method is based on interactive robot control which requires immediate response of robot motions but could deal with unstructured visual patterns, which rejects all datasets which only store offline and static images with well-known visual patterns. However, the settings and results presented could be used as a baseline to be compared to by other future research works.

B. Practicability

First, our robot–camera calibration approach allows very simple set-up. It only requires stable observation of a single feature point with unknown position, which is theoretically the most relaxed assumption for image feedback and is easy to generate (even a natural stain suffices as in our experiments). No initialization step is required as well, compared to the recent works in [38], [56], and [10] that uses a Perspective-n-Point (PnP) solver to compute an initial guess. Second, our method does not require tedious data acquisition process. The abovementioned result in Figs. 16 and 17 is achieved by assigning $n_R = 5$ and $n_t = 20$, where a total time of 1 min (at 30 fps) suffices the acquisition and computing process, which is faster than many existing works (e.g., 40 mins in [41]). No large-scale data collection is required as well, compared to the 256 groups of data required for [39] targeting similar assumptions. Moreover, the motion space of the end-effector during data acquisition could be rigorously restricted using out IP-based control scheme with no calibration objects. Although we set the camera-feature distance in our experiment as $\sim$100 mm, the capability of our algorithm using a single feature point does not impose any theoretical limit of such distance, as long as the feature could be focused by the camera for stable image-based tracking.

The calibration process is autonomous and the available motion space could be manually defined by users via ϕ, which shows great intrinsic potential for application in complex working environment, comparing to the approaches requiring calibration objects. We currently do not address the robot motions reaching limits and/or singularities. However, as the motion only covers a very small volume proportion ($\sim 3\%$ of the dVRK's workspace) and the fact that the reference $\mathbf{p}_0$ is manually selected, this naturally avoids the possibility to meet so.

C. Generality

Besides the eye-to-hand set-up as demonstrated in this work, our method is easily extendable to solving eye-in-hand calibration and robot-to-robot calibration. Foe the eye-in-hand case, one could directly apply the end-effector motion depicted by (5) to the robot that carries the camera. The feature point could be selected as a static point in the world. The IP-based control scheme again aims to settle the image error (10) from the precomputed l_r, until $\mathbf{B}$ and $\mathbf{C}$ are solved. The rest of the computation remains identical to the current eye-to-hand model but only with t_m representing the feature's position with respect to the robot base frame instead of the end-effector frame.

For the robot-to-robot calibration, we still only require one feature point for each end-effector to generate image feedback. To construct n_r, one could directly connect the two feature points observable in the image upon the robots' initial configuration to form n_r. Then, assuming robot I and robot II, each robot is independently controlled to collect respective $\mathbf{B}_1, \mathbf{C}_1$ and $\mathbf{B}_2, \mathbf{C}_2$ using the same pipeline as in this article. The rotation between robot I and robot II could then be solved as in Section V-B by eliminating the camera term.

IX. Conclusion

In this article, we have proposed a new approach to solve robot–camera calibration under tightly constrained environment. The approach considered a single (position-unknown) feature point in a user-defined restricted motion space for data acquisition. We then leveraged the concept of IP to enlarge the acquirable feedback data, and proposed the IFP, which is adjusted via a novel IP-based control scheme to acquire spatial relationship between the robot and the camera. Simulations have shown the capability of our algorithm to calibrate the robot–camera relationship on different types of robot accurately and robustly. In experiments, the results demonstrated the superiority of our algorithm compared to existing methods on the 3-D end-effector positioning error and 2-D back-projection error. Computation consistency is not significantly affected compared to approaches using calibration objects. Our method does not require calibration objects nor large-scale data acquisition, and

could be extended to solving robot–camera and robot–robot calibration.

Aiming for future work, we plan to further enhance the performance of our algorithm across more complex applications. At current stage, we take all the model errors and/or measurement noises into the estimation phase. This could provide robust results under a considerable scale of data, but is hard to further improve the calibration accuracy, which is one limitation in this work. As different sources of errors affect the results differently, using data-driven algorithms to "learn" the error distribution (e.g., robot joint errors) might facilitate better accuracy under tight environment. It is also promising to apply such method on continuum robots and soft robots in the future whose kinematics error is not negligible, and to online update the calibration results if on-the-fly recalibration is required in complex robotic applications.

APPENDIX A
PROOF OF THEOREM 2

We first combine (4), (5), and (22) to obtain the following:

$$\begin{aligned}\mathbf{0}_{6\times1} =& \begin{bmatrix}\mathbf{R}_{\mathbf{q}_0}\dfrac{\partial \mathbf{v}}{\partial \boldsymbol{\Psi}}(\boldsymbol{\Psi}(t))\\ \mathbf{0}_{3\times6}\end{bmatrix}\\ &+\mathbf{J}(\cdot)\mathbf{J}^{-1}(\cdot)\boldsymbol{\Gamma}\begin{bmatrix}\mathbf{r}(\mathbf{q}(t))-\mathbf{x}_{\mathbf{q}_0}-\phi\mathbf{R}_{\mathbf{q}_0}\mathbf{v}(\boldsymbol{\Psi}(t))\\ \mathbf{0}_{3\times6}\end{bmatrix}\\ &+\phi\mathbf{J}(\cdot)\mathbf{J}^{-1}(\cdot)\begin{bmatrix}\mathbf{R}_{\mathbf{q}_0}\dfrac{\partial \mathbf{v}}{\partial \boldsymbol{\Psi}}(\boldsymbol{\Psi}(t))\dot{\boldsymbol{\Psi}}(t)\\ \mathbf{0}_{3\times6}\end{bmatrix}\\ \mathbf{0}_{6\times1} =& \begin{bmatrix}\dot{\mathbf{r}}(\mathbf{q}(t))+\phi\mathbf{R}_{\mathbf{q}_0}\dfrac{\partial \mathbf{v}}{\partial \boldsymbol{\Psi}}(\boldsymbol{\Psi}(t))\dot{\boldsymbol{\Psi}}(t)\\ \mathbf{0}_{3\times6}+\mathbf{0}_{3\times6}\end{bmatrix}\\ &+\boldsymbol{\Gamma}\begin{bmatrix}\mathbf{r}(t)+\mathbf{R}_{\mathbf{q}_0}\bar{\mathbf{t}}_m-\mathbf{x}_{\mathbf{q}_0}-\phi\mathbf{R}_{\mathbf{q}_0}\mathbf{v}(\boldsymbol{\Psi}(t))-\mathbf{R}_{\mathbf{q}_0}\bar{\mathbf{t}}_m\\ \mathbf{w}_0-\mathbf{w}_0\end{bmatrix}.\end{aligned} \tag{35}$$

As one can compute the robot-actuated position of the feature point as $\mathbf{m}(\mathbf{q}(t))=\mathbf{r}(t)+\mathbf{R}_{\mathbf{q}_0}\bar{\mathbf{t}}_m$, incorporating (7) could, thus, lead the first three rows to the following dynamics:

$$\dot{\mathbf{m}}_\zeta(\cdot)=\dot{\mathbf{m}}(\mathbf{q}(t))+\boldsymbol{\Gamma}(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot)). \tag{36}$$

To evaluate the performance of (36), we define the following Lyapnuov function as

$$V=\frac{1}{2}\left(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot)\right)^\intercal\left(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot)\right) \tag{37}$$

then, the derivative of (37) by considering (36) could be derived into the following form:

$$\begin{aligned}\dot{V}&=(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot))^\intercal(\dot{\mathbf{m}}(\mathbf{q}(t))-\dot{\mathbf{m}}_\zeta(\cdot))\\ &=(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot))^\intercal(\dot{\mathbf{m}}(\mathbf{q}(t))-\dot{\mathbf{m}}(\mathbf{q}(t))\\ &\quad-\boldsymbol{\Gamma}(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot))\\ &=-(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot))^\intercal\boldsymbol{\Gamma}(\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot))\end{aligned} \tag{38}$$

which indicates a negative semi-definite function. Meanwhile, as $\mathbf{m}(\mathbf{q}(t))$ and $\mathbf{m}_\zeta(\cdot)$ are robot-actuating states and to be smoothly planned during each IP control process (subject to s), they are obviously bounded which leads $\dot{V}$ to be uniformly continuous. Thus, the evolution of $\mathbf{m}(\mathbf{q}(t))-\mathbf{m}_\zeta(\cdot)$ is asymptotically minimized to $\mathbf{0}$ based on the Barbalat's Lemma [47]. The following statement:

$$\begin{aligned}\lim_{t\to\infty}||&\mathbf{r}(t)+\mathbf{R}_{\mathbf{q}_0}\bar{\mathbf{t}}_m\\ &-\underbrace{(\mathbf{x}_{\mathbf{q}_0}+\phi\mathbf{R}_{\mathbf{q}_0}\mathbf{v}(\boldsymbol{\Psi}(t))+\mathbf{R}_{\mathbf{q}_0}\bar{\mathbf{t}}_m)}_{\mathbf{m}_{\zeta,\mathbf{q}_0}(\boldsymbol{\Psi}(t))}||=0\end{aligned} \tag{39}$$

is then satisfied.

REFERENCES

[1] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, "A comparative review of hand-eye calibration techniques for vision guided robots," *IEEE Access*, vol. 9, pp. 113143–113155, 2021.

[2] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, "Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 417–436, Feb. 2023.

[3] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.

[4] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving AX = XB on the Euclidean group," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 717–721, Oct. 1994.

[5] I. Ali, O. Suominen, A. Gotchev, and E. R. Morales, "Methods for simultaneous robot-world-hand–eye calibration: A comparative study," *Sensors*, vol. 19, no. 12, 2019, Art. no. 2837.

[6] M. Lai, C. Shan, and P. H. de With, "Hand-eye camera calibration with an optical tracking system," in *Proc. 12th Int. Conf. Distrib. Smart Cameras*, 2018, pp. 1–6.

[7] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: A current perspective," *Ann. Surg.*, vol. 239, no. 1, 2004, Art. no. 14.

[8] C. D'Ettorre et al., "Accelerating surgical robotics research: A review of 10 years with the da Vinci research kit," *IEEE Robot. Autom. Mag.*, vol. 28, no. 4, pp. 56–78, Dec. 2021.

[9] A. Edsinger, "Robot manipulation in human environments," Ph.D. thesis, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA, 2007.

[10] F. Richter, J. Lu, R. K. Orosco, and M. C. Yip, "Robotic tool tracking under partially visible kinematic chain: A unified approach," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1653–1670, Jun. 2022.

[11] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale, "Online self-calibration for robotic systems," *Int. J. Robot. Res.*, vol. 35, no. 4, pp. 357–380, 2016.

[12] F. Zhong, Z. Wang, W. Chen, K. He, Y. Wang, and Y.-H. Liu, "Hand-eye calibration of surgical instrument for robotic surgery using interactive manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1540–1547, Apr. 2020.

[13] E. Pedrosa, M. Oliveira, N. Lau, and V. Santos, "A general approach to hand–eye calibration through the optimization of atomic transformations," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1619–1633, Oct. 2021.

[14] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX= XB," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, Feb. 1989.

[15] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[16] H. Zhuang, K. Wang, and Z. S. Roth, "Simultaneous calibration of a robot and a hand-mounted camera," *IEEE Trans. Robot. Autom.*, vol. 11, no. 5, pp. 649–660, Oct. 1995.

[17] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, 1999.

[18] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and robot-world calibration by global polynomial optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3157–3164.

[19] H. Zuang and Y. C. Shiu, "A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 4, pp. 1168–1175, Jul./Aug. 1993.

[20] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 617–622, Aug. 1998.

[21] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 4647–4653.

[22] Z. Zhao, "Hand-eye calibration using convex optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2947–2952.

[23] A. Tabb and K. M. Ahmad Yousef, "Solving the robot-world hand-eye (s) calibration problem with iterative methods," *Mach. Vis. Appl.*, vol. 28, no. 5, pp. 569–590, 2017.

[24] Q. Zhan and X. Wang, "Hand–eye calibration and positioning for a robot drilling system," *Int. J. Adv. Manuf. Technol.*, vol. 61, no. 5, pp. 691–701, 2012.

[25] J. Zhang, F. Shi, and Y. Liu, "An adaptive selection of motion for online hand-eye calibration," in *Proc. Australas. Joint Conf. Artif. Intell.*, 2005, pp. 520–529.

[26] J. Schmidt and H. Niemann, "Data selection for hand-eye calibration: A vector quantization approach," *Int. J. Robot. Res.*, vol. 27, no. 9, pp. 1027–1053, 2008.

[27] G. Wang et al., "Simultaneous calibration of multicoordinates for a dual-robot system by solving the AXB= YCZ problem," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1172–1185, Aug. 2021.

[28] K. Stepanova, J. Rozlivek, F. Puciow, P. Krsek, T. Pajdla, and M. Hoffmann, "Automatic self-contained calibration of an industrial dual-arm robot with cameras using self-contact, planar constraints, and self-observation," *Robot. Comput.- Integr. Manuf.*, vol. 73, 2022, Art. no. 102250.

[29] J. Jiang, X. Luo, Q. Luo, L. Qiao, and M. Li, "An overview of hand-eye calibration," *Int. J. Adv. Manuf. Technol.*, vol. 119, no. 1, pp. 77–97, 2022.

[30] 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, Jun. 2020.

[31] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *Proc. IEEE 2nd Int. Conf. 3-D Digit. Imag. Model.*, 1999, pp. 430–436.

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

[33] J. Schmidt, F. Vogt, and H. Niemann, "Calibration–free hand–eye calibration: A structure–from–motion approach," in *Proc. Joint Pattern Recognit. Symp.*, 2005, pp. 67–74.

[34] K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1021–1028, Apr. 2019.

[35] K. Pauwels and D. Kragic, "Integrated on-line robot-camera calibration and object pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 2332–2339.

[36] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," in *Proc. 15th Int. Conf. Adv. Robot.*, 2011, pp. 67–74.

[37] J.-S. Hu and Y.-J. Chang, "Automatic calibration of hand–eye–workspace and camera using hand-mounted line laser," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 6, pp. 1778–1786, Dec. 2012.

[38] M. Ye, L. Zhang, S. Giannarou, and G.-Z. Yang, "Real-time 3D tracking of articulated tools for robotic surgery," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2016, pp. 386–394.

[39] Z. Wang et al., "Vision-based calibration of dual RCM-based robot arms in human-robot collaborative minimally invasive surgery," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 672–679, Apr. 2018.

[40] K. Pachtrachai, M. Allan, V. Pawar, S. Hailes, and D. Stoyanov, "Hand-eye calibration for robotic assisted minimally invasive surgery without a calibration object," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2485–2491.

[41] O. Özgüner et al., "Camera-robot calibration for the da vinci robotic surgery system," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 2154–2161, Oct. 2020.

[42] A. Roberti, N. Piccinelli, D. Meli, R. Muradore, and P. Fiorini, "Improving rigid 3-D calibration for robotic surgery," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 4, pp. 569–573, Nov. 2020.

[43] J. Bohg et al., "Interactive perception: Leveraging action in perception and perception in action," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1273–1291, Dec. 2017.

[44] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz, "Interactive segmentation, tracking, and kinematic modeling of unknown 3D articulated objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5003–5010.

[45] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 272–277.

[46] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[47] J.-J. E. Slotine et al. *Applied Nonlinear Control*, vol. 199. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991.

[48] O. Rodrigues, "Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire," *J. Math. Pures Appl*, vol. 5, pp. 380–400, 1840.

[49] G. Piovan and F. Bullo, "On coordinate-free rotation decomposition: Euler angles about arbitrary axes," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 728–733, Jun. 2012.

[50] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Math. Program.*, vol. 89, no. 1, pp. 149–185, 2000.

[51] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci surgical system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 6434–6439.

[52] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "A V-REP simulator for the da vinci research kit robotic platform," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatron.*, 2018, pp. 1056–1061.

[53] P. Escobar and T. Falcone, *Atlas of Single-Port, Laparoscopic, and Robotic Surgery*, Berlin, Germany: Springer, 2014.

[54] F. Cursi, W. Bai, E. M. Yeatman, and P. Kormushev, "Model learning with backlash compensation for a tendon-driven surgical robot," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7958–7965, Jul. 2022.

[55] Z. Zhang, L. Zhang, and G.-Z. Yang, "A computationally efficient method for hand–eye calibration," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 10, pp. 1775–1787, 2017.

[56] M. Allan, S. Ourselin, D. J. Hawkes, J. D. Kelly, and D. Stoyanov, "3-D pose estimation of articulated instruments in robotic minimally invasive surgery," *IEEE Trans. Med. Imag.*, vol. 37, no. 5, pp. 1204–1213, May 2018.

Fangxun Zhong (Member, IEEE) received the B.Eng. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2014 and the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong, China, in 2021.

He is currently a Postdoctoral Fellow with the T Stone Robotics Institute and Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong. His research interests include surgery autonomy, medical robotics, robot control in complex environment, and deep learning in medical applications.

Bin Li (Student Member, IEEE) received the bachelor's degree (first honor) from the Department of Mechatronics Engineering, Harbin Institute of Technology, Harbin, China, in 2016, and the master's degree (first honor) in mechanical engineering from Shanghai Jiao Tong University, China, in 2019. He is currently working toward the Ph.D. degree with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong.

His research interests include surgical autonomy, computer vision, and imitation learning.

Wei Chen (Student Member, IEEE) received the B.E. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2012 and the M.S. degree in mechanical and automation engineering in 2021 from The Chinese University of Hong Kong, Hong Kong, where he is currently working toward the Ph.D. degree with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong SAR.

His research interests include medical robots and soft robotics.

Yun-Hui Liu (Fellow, IEEE) received the Ph.D. degree in applied mathematics and information physics from the University of Tokyo, Tokyo, Japan, in 1992.

After working with the Electrotechnical Laboratory of Japan as a Research Scientist, he joined The Chinese University of Hong Kong (CUHK), Hong Kong, in 1995 and is currently Choh-Ming Li Professor of Mechanical and Automation Engineering and the Director of the T Stone Robotics Institute. He is also the Director/CEO of Hong Kong Centre for Logistics Robotics sponsored by the InnoHK Programme of the HKSAR Government. He is an Adjunct Professor with the State Key Lab of Robotics Technology and System, Harbin Institute of Technology, Harbin, China. He has authored or coauthored more than 500 papers in refereed journals and refereed conference proceedings and was listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. His research interests include visual servoing, logistics robotics, medical robotics, multifingered grasping, mobile robots, and machine intelligence.

Dr. Liu was the recipient of numerous research awards from international journals and international conferences in robotics and automation and government agencies. He was the Editor-in-Chief of Robotics and Biomimetics and was an Associate Editor for IEEE TRANSACTION ON ROBOTICS AND AUTOMATION and General Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.