Robotica (2023), **41**, pp. 2648–2667
doi:10.1017/S0263574723000589

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

A low-cost indoor positioning system based on data-driven modeling for robotics research and education

Junlin Ou[1], Seong Hyeon Hong[2], Tristan Kyzer[1], Haizhou Yang[3], Xianlian Zhou[4] and Yi Wang[1]

[1]Department of Mechanical Engineering, University of South Carolina, Columbia, SC 29208, USA, [2]Department of Mechanical and Civil Engineering, Florida Institute of Technology, Melbourne, FL 32901, USA, [3]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA, and [4]Department of Department of Biomedical Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA
Corresponding author: Yi Wang; Email: yiwang@cec.sc.edu

Received: 31 December 2022; **Revised:** 21 March 2023; **Accepted:** 11 April 2023; **First published online:** 11 May 2023

Keywords: ArUco marker; OpenCV; indoor positioning system; image processing; data-driven model

Abstract

This paper presents a low-cost, accurate indoor positioning system that integrates image acquisition and processing and data-driven modeling algorithms for robotics research and education. Multiple overhead cameras are used to obtain normalized image coordinates of ArUco markers, and a new procedure is developed to convert them to the camera coordinate frame. Various data-driven models are proposed to establish a mapping relationship between the camera and the world coordinates. One hundred fifty data pairs in the camera and world coordinates are generated by measuring the ArUco marker at different locations and then used to train and test the data-driven models. With the model, the world coordinate values of the ArUco marker and its robot carrier can be determined in real time. Through comparison, it is found that a straightforward polynomial regression outperforms the other methods and achieves a positioning accuracy of about 1.5 cm. Experiments are also carried out to evaluate its feasibility for use in robot control. The developed system (both hardware and algorithms) is shared as an open source and is anticipated to contribute to robotic studies and education in resource-limited environments and underdeveloped regions.

1. Introduction

Object positioning techniques [1, 2], particularly those low-cost but accurate, have gained significant traction in robotics research and education. Several of them have found widespread applications in the real world, such as location-based service and navigation. Global Positioning System (GPS) is one of the greatest revolutions in the localization application, and it can provide positioning information for almost all receivers on earth. However, it is not entirely amenable to indoor environments because the satellite signals can be blocked significantly by the walls of building construction [2]. Furthermore, the GPS accuracy (namely, the distance error between the ground truth and the reported position) of low-cost sensors is at the level of ∼meters, and therefore, it cannot satisfy the requirements of many indoor applications. Existing indoor positioning methods can be classified into three categories, including pedestrian dead reckoning (PDR) [3–5], communication technology [6–8], and computer vision [9–12], and each has its advantages and drawbacks.

PDR estimates the object's position through its past positions and the measurement data from magnetometers, gyroscopes, accelerometers, and others [12]. PDR is still a popular option for indoor localization and is often implemented through smartphones. However, its positioning error is generally high and accumulates as the object moves away from its initial location. Kuang et al. [13] developed a PDR algorithm using a quasi-static attitude, a magnetic field vector, and a gravity vector. In addition, the motion constraint and gait models are applied to make PDR algorithm more robust. Experiments were performed to verify that the proposed algorithm improved positioning accuracy over an existing PDR

method. The mean positioning error could be up to 2.08 m. Wang et al. [14] proposed a motion-mode recognition-based PDR using smartphones. The decision-tree and support vector machine (SVM) algorithms were used to recognize phone poses and movement states, which improved localization accuracy. It was reported that the mean error of different phone poses was at least 1.38 m in a trajectory of 164 m. Liu et al. [15] presented an enhanced PDR algorithm with the support of digital terrestrial multimedia broadcasting (DTMB) signals. Furthermore, the extended Kalman filter algorithm was used to fuse the information of the Doppler speed and range, and pedestrian walking speed and heading from DTMB signals and PDR, further boosting the performance. Compared with the native PDR, 95% of positioning errors of the enhanced PDR algorithm are much smaller and less than 3.94 m. However, the positioning accuracies of PDRs (including those with enhancement algorithms) generally are insufficient for the control or obstacle avoidance of mobile robots in the indoor environment.

Communication-based approaches include ultra-wideband (UWB), Bluetooth, Wi-Fi, radio frequency identification, and visible light communication. Compared to the PDR methods, they can provide more accurate positioning information, and their positioning errors do not change as the distance between the object and the initial location varies. Ruiz et al. [16] compared the positioning performance of three commercial UWB positioning systems, BeSpoon, DecaWave, and Ubisense. It was found in experiments that DecaWave outperformed BeSpoon in accuracy and both exceeded the Ubisense. Within the same testing environment, the mean positioning errors of BeSpoon, DecaWave, and Ubisense were 0.71, 0.49, and 1.93 m, respectively. Sthapit et al. [17] proposed a Bluetooth-based indoor positioning method using machine learning. Sample data from the Bluetooth device of low energy consumption were used to train a machine learning model. Then, experiments were carried out to evaluate the machine learning algorithm, and the average location error was found to be 50 cm. Increasing the sample size could further reduce the localization error. Han et al. [18] presented a new WiFi-based approach along with an algorithm for indoor positioning. Their approach achieved a higher accuracy than the traditional WKNN (weighted K-nearest neighbor) algorithm. Specifically, the positioning errors of the proposed and the traditional WKNN algorithms are 0.25 and 0.37 m, respectively.

Computer vision-based methods for indoor positioning localize objects by analyzing contents in imagery or video data, and the widely used algorithms include clustering, matching, feature extraction, and deep learning. The accuracy of computer vision-based approaches usually is higher than that of their communication-based counterparts. However, the range of computer vision-based methods is limited since the view of a single camera is restricted, and this issue can be resolved by combining multiple cameras. Jia et al. [19] proposed a deep multipatch network-based image deblurring algorithm to enhance accuracy in indoor visual positioning by eliminating the blurry effect and improving the image quality, which achieved an average positioning accuracy of 8.65 cm in an office environment and outperformed other methods, such as continuous indoor visual localization and indoor image-based localization method. In ref. [20], an indoor visual positioning method utilizing image features was proposed. The image features were extracted from depth information and RGB channels in the images. Then, a bundle adjustment method and an efficient perspective n-point method were applied to implement indoor positioning. The proposed method was verified in the real environment, and its root mean square error could reach 0.129 m. Li et al. [21] presented an indoor visible light positioning system with optical camera communication. After capturing image data using the camera in a smartphone, a novel perspective-n-point problem algorithm was used to estimate the smartphone's position. The proposed system was verified through experiments and obtained the mean position error of 4.81 cm while the object was placed at a height of 50 cm. Lastly, high-quality computer vision-based localization systems are also commercially available, like OptiTrack camera systems and Vicon systems. They offer even higher positioning accuracy at the level of millimeters. However, such positioning systems typically use a large number of cameras from different perspective angles and need complicated installation, leading to high costs (ten thousand or several hundred thousand dollars depending on the quality). Hence, they are not affordable for robotics research and education in resource-limited environments or geographically underdeveloped regions. Therefore, there is a critical need for an indoor position system with an excellent balance between cost and accuracy because extremely high accuracy and precision may not

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

be necessary for entry- or intermediate-level robotics research and education purposes. The specific challenge is how to retain desirable positioning accuracy (a few centimeters) and precision ($\sim$ 1 cm) while keeping the equipment and the installation cost low (e.g., < \$300). Such a system would not only generate positioning data to meet the need for research and educational programs but also represent a financially viable solution for advocating these activities.

Considering the accuracy and cost requirements surveyed above, we propose a cost-effective method and system for accurate indoor positioning for robotics research and education. The underlying idea is to utilize multiple low-cost cameras to acquire images on an ArUco marker attached to mobile robots. The cameras are arranged in a plane to significantly enlarge the view range for practical use and facilitate system installation. In addition, a new process that combines computer vision techniques to extract camera coordinates and data-driven models to establish a quantitative mapping between the camera and the world coordinates is also developed. The rationale for employing fiducial markers is that they have proven very effective in improving object positioning because of their highly distinguishable patterns [22]. ARTag, AprilTag, ArUco, and STag markers are the most widely used fiducial markers, and among them, the ArUco maker requires the lowest computation cost while maintaining salient positioning accuracy. Hence, it emerges as the best option for the proposed system [23]. For example, the positioning accuracy of around 10 cm was achieved with the ArUco marker in experiments [2], although the positioning range is somewhat limited due to the use of only one camera.

The novelty and contributions of the present work are summarized as follows:

1. A novel and holistic solution for low-cost (< \$300) but accurate (error $\sim$ 1.5 cm) indoor positioning for robotics research and education is proposed. The system integrates hardware and algorithm development and software–hardware interfacing for automated image acquisition and processing, computing, and positioning, all in real time.
2. The present effort aims to push the limit of positioning accuracy by improving numerical algorithms and software while minimizing the overall cost of hardware and installation. Therefore, a new algorithmic pipeline is developed to combine computer vision and data-driven models, which converts the 2D images obtained by low-cost cameras to 3D world coordinates. Multiple algorithms and models are available for use in different scenarios.
3. All algorithms in Python and Matlab, including real-time image streaming from camera hardware, image processing and computer vision, and data-driven models, as well as experimental data for model training and testing presented in this work, are shared as an open source in the public domain.[1]

It should be emphasized that the goal of the present effort is not to replace the high-quality indoor imaging systems of commercial grades for sophisticated robotics applications. Instead, it aims to realize a cost-effective system to meet basic research and education needs in resource-deficient environments.

The remainder of this work is organized as follows. In Section 2, the indoor positioning system and multiple approaches for world coordinate estimation are described in detail. Section 3 introduces the experimental setup for data collection and evaluation of the positioning system. Experimental results and performance characterization are discussed in Section 4. Section 5 concludes the paper with a brief summary and potential future work.

2. Positioning method and system

In this section, the positioning method and system is introduced, and the data-driven models used to calibrate and improve the world coordinate estimation are also described in detail. As shown in Fig. 1, the entire pipeline consists of two stages, offline training and online testing/utilization. During the offline training stage, multiple overhead cameras will be used to capture the image of the floorboard (labeled "*a*"

[1]The open source library is available at https://github.com/iMSEL-USC/Indoor-Positioning-System-iMSEL.

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

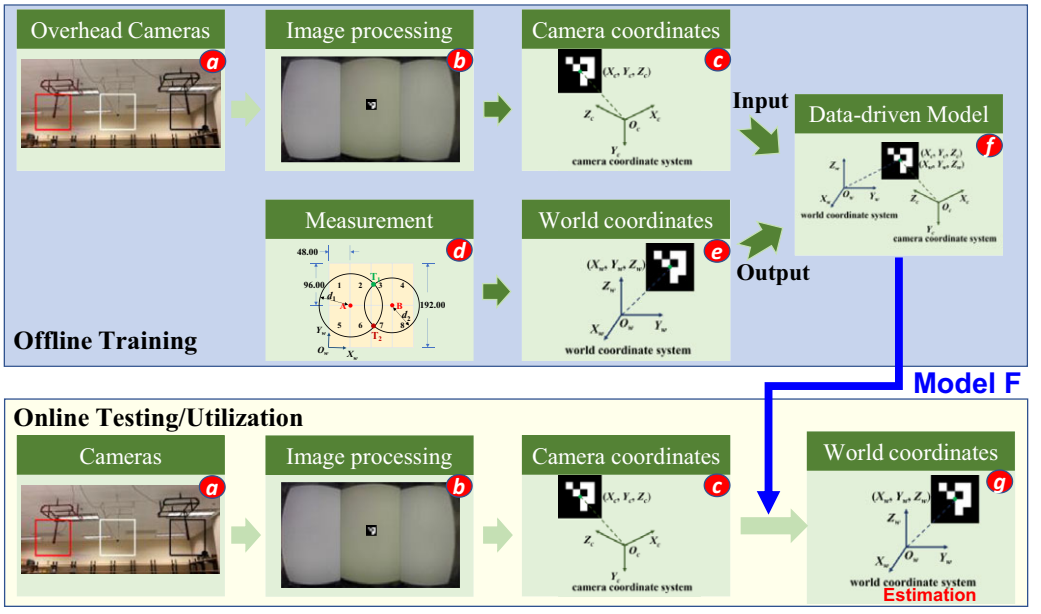

Figure 1. *The method and workflow of the developed low-cost indoor positioning system.*

in Fig. 1), and the ArUco marker will be placed at specified locations and heights. Then, the image is processed to obtain the normalized image coordinates $(u, v, 1)$ of the ArUco marker using OpenCV (labeled "*b*"). The normalized image coordinates are then converted to the camera coordinates (X_c, Y_c, Z_c) using a new algorithm developed in this work (labeled "*c*"). Meanwhile, through manual measurement (labeled "*d*"), the ground truth values of the world coordinates (X_W, Y_W, Z_W) of the ArUco marker are attained (labeled "*e*"). These steps, that is, image acquisition and processing and computing camera coordinates and true world coordinate values, are repeated multiple times by placing the ArUco maker at different locations, which will generate sufficient data pairs of the camera and world coordinates. A data-driven model (labeled "*f*"), such as rigid transformation, polynomial regression, artificial neural network, and Kriging will be trained to establish a mapping relationship $\boldsymbol{F}$ between the camera coordinate (as input) and world coordinates (as output) in the previous steps. During the online testing/utilization stage, the mobile robot carrying the ArUco marker will be captured by the overhead cameras, and the image will be processed to produce the normalized image coordinates $(u, \nu, 1)$ and camera coordinates (X_c, Y_c, Z_c) following the same procedure above (i.e., "*a*", "*b*", and "*c*" steps). Differently, the camera coordinate will be entered as the input to the data-driven model $\boldsymbol{F}$ trained in the offline stage to immediately estimate the world coordinates $(\hat{X}_w, \hat{Y}_w, \hat{Z}_w)$ (labeled "*g*"). In other words, the trained data-driven model is utilized (as shown by the blue arrow) in the online stage.

2.1. Camera coordinate frame

As shown in Fig. 2, (X_c, Y_c, Z_c) and (X_w, Y_w, Z_w) represent the camera and world coordinate frames, respectively. For a point located at the center of the ArUco marker, its camera and world coordinate values are (X_{c0}, Y_{c0}, Z_{c0}) and (X_{w0}, Y_{w0}, Z_{w0}), respectively. The marker has a square shape and four corner points, which are denoted by (X_{c1}, Y_{c1}, Z_{c1}), (X_{c2}, Y_{c2}, Z_{c2}), (X_{c3}, Y_{c3}, Z_{c3}), and (X_{c4}, Y_{c4}, Z_{c4}) in the camera coordinate frame. In our lab experiments, a process (detailed in Section 3) is developed to align the cameras almost parallel to the floorboard. The ArUco marker is placed flat on the floorboard and has a small size; therefore, it is valid to assume $Z_{c1} = Z_{c2} = Z_{c3} = Z_{c4}$.

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

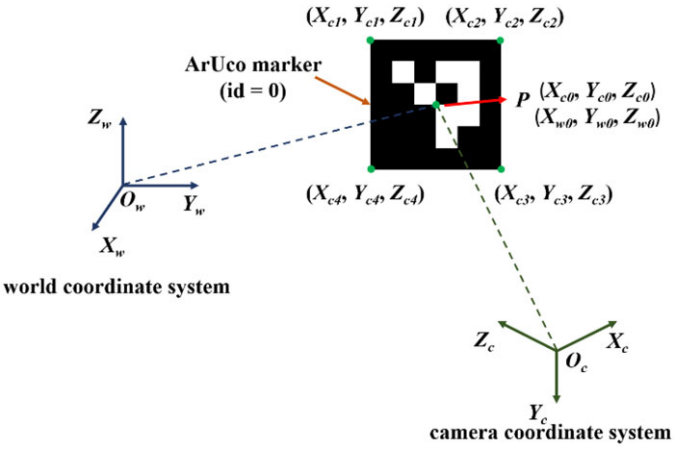

Figure 2. *Camera and world coordinate frames.*

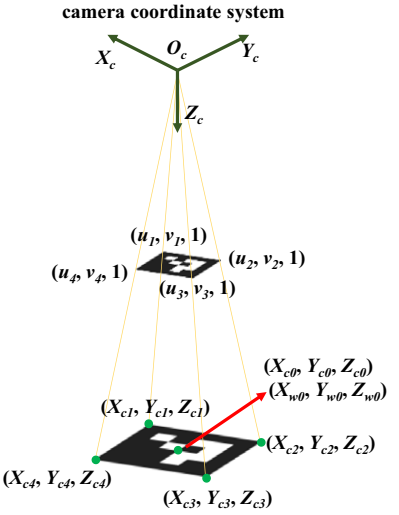

Figure 3. *Normalized image coordinates and camera coordinates of the ArUco marker.*

The length L of the four edges of the ArUco marker in the camera coordinate is the same and can be expressed as

$$L = \sqrt{(X_{cj} - X_{ci})^2 + (Y_{cj} - Y_{ci})^2 + (Z_{cj} - Z_{ci})^2} \approx \sqrt{(X_{cj} - X_{ci})^2 + (Y_{cj} - Y_{ci})^2}$$
$$= \frac{1}{4} \sum_{i=1}^{4} \sqrt{(X_{cj} - X_{ci})^2 + (Y_{cj} - Y_{ci})^2} \tag{1}$$

where $i = 1, 2, 3,$ and 4, $j = \mathrm{mod}(i, 4) + 1$, and 'mod' operation denotes the remainder after division. Usually, the values of the corners (X_{ci}, Y_{ci}, Z_{ci}) in the camera coordinate frame are unknown. However, we can find them from their normalized coordinates $(u, v, 1)$ using OpenCV undistortPoints() function, where $u = X_c/Z_c$ and $v = Y_c/Z_c$. This function corrects lens distortion and normalizes the coordinates of detected points. According to the triangle similarity Theorems (Fig. 3), Eq. (2) can also be expressed as

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

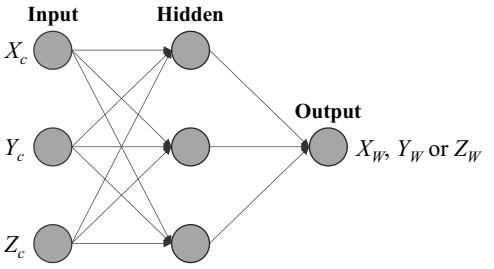

Figure 4. *The neural network model used in this study.*

$$L = \frac{Z_{c0}}{4} \sum_{i=1}^{4} \sqrt{\left(u_j - u_i\right)^2 + \left(v_j - v_i\right)^2} \tag{2}$$

where Z_{c0} is the value of Z_c at the marker center, and $Z_{c0} = Z_{c1} = Z_{c2} = Z_{c3} = Z_{c4}$ because the marker is small, flat, and almost parallel to the camera. In this work, the marker length L is measured manually, which allows us to calculate Z_{c0} of the marker by

$$Z_{c0} = \frac{4L}{\sum_{i=1}^{4} \sqrt{\left(u_j - u_i\right)^2 + \left(v_j - v_i\right)^2}} \tag{3}$$

As the marker is a square, the x and y coordinates of its center, that is, X_{c0} and Y_{c0}, can be written as

$$X_{c0} = \sum_{i=1}^{4} X_{ci} = \frac{Z_{c0}}{4} \sum_{i=1}^{4} u_i. \tag{4}$$

$$Y_{c0} = \sum_{i=1}^{4} Y_{ci} = \frac{Z_{c0}}{4} \sum_{i=1}^{4} v_i. \tag{5}$$

Thus, the camera coordinate values of the marker's center, that is, (X_{c0}, Y_{c0}, Z_{c0}) can be completely determined by Eqs. (3)-(5).

2.2. World coordinate frame

The next step is to transform the ArUco marker from the camera coordinate frame to the world coordinate frame for localization in the real environment. To establish the transformation relationship $\boldsymbol{F}$ between them, that is, $(X_w, Y_w, Z_w) = \boldsymbol{F}(X_c, Y_c, Z_c)$, the true value of the ArUco marker in the world coordinate frame is needed and can be manually measured by treating one location in the real environment as the origin. Note that the location of the ArUco marker in the camera coordinate is now known following the procedure in Section 2.1. The ArUco marker is placed at multiple locations, and the measurement is repeated accordingly, which yields a dataset containing many pairs of (X_c, Y_c, Z_c) and (X_w, Y_w, Z_w), each corresponding to one marker location. The dataset is then split into two groups, respectively, for training and testing of model $\boldsymbol{F}$.

The transformation relationship $\boldsymbol{F}$ can be identified by various data training/learning approaches, such as rigid transformation, polynomial regression, Kriging interpolation, machine learning, and others, which are described in detail below.

2.2.1. Rigid transformation

Rigid transformation is the most widely used approach to estimate mapping between the camera and the world coordinates. It is a kind of transformation that does not change the Euclidean distance of every

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

pair of points, such as translations and rotations (that are considered in this work). It is mathematically expressed as

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{\Gamma}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{6}$$

where $\boldsymbol{R}$ and $\boldsymbol{\Gamma}$ are the rotation matrix and translation matrix, respectively. Given the dataset in the camera and world coordinate frames above, $\boldsymbol{R}$ and $\boldsymbol{\Gamma}$ can be computed by singular value decomposition (SVD) [24, 25]. The data pairs in the camera frame $\boldsymbol{X}$ and in the world frame $\boldsymbol{Y}$ are

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}^{(1)} \\ \vdots \\ \boldsymbol{x}^{(n)} \end{pmatrix}, \boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(n)} \end{pmatrix} \tag{7}$$

where $\mathbf{X} \in \mathfrak{R}^{n\times3}$ is the matrix composed of n observations for the three measured input quantities (X_c, Y_c, and Z_c), that is, each entry $\boldsymbol{x}^{(i)} = (X_c^{(i)}, Y_c^{(i)}, Z_c^{(i)})$ is the ith observation and $0 \le i \le n$; $\mathbf{Y} \in \mathfrak{R}^{n\times3}$ is the matrix containing n measurements for the three output quantities (X_w, Y_w, or Z_w), and each entry $\boldsymbol{y}^{(i)} = (X_w^{(i)}, Y_w^{(i)}, Z_w^{(i)})$. The centroids of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are calculated by

$$\overline{\boldsymbol{X}} = \frac{1}{n}\sum\nolimits_{i=1}^{n} \boldsymbol{x}^{(i)}, \text{ and } \overline{\boldsymbol{Y}} = \frac{1}{n}\sum\nolimits_{i=1}^{n} \boldsymbol{y}^{(i)} \tag{8}$$

Note that $\overline{\boldsymbol{X}} \in \mathfrak{R}^{1\times3}$ and $\overline{\boldsymbol{Y}} \in \mathfrak{R}^{1\times3}$ are row vectors in 3D. Then, the covariance matrix $\boldsymbol{H}$ [24, 25] is defined as

$$\boldsymbol{H} = (\boldsymbol{X} - \overline{\boldsymbol{X}})^T(\boldsymbol{Y} - \overline{\boldsymbol{Y}}) \tag{9}$$

$(\boldsymbol{X} - \overline{\boldsymbol{X}})$ is an operation that each row of $\boldsymbol{X}$ is subtracted by $\overline{\boldsymbol{X}}$. Then, $\boldsymbol{H}$ is decomposed by SVD to produce $\boldsymbol{U}$ and $\boldsymbol{V}$:

$$[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = SVD(\boldsymbol{H}) \tag{10}$$

Thus, the $\boldsymbol{R}$ and $\boldsymbol{\Gamma}$ can be obtained

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^T \tag{11}$$

$$\boldsymbol{\Gamma} = \overline{\boldsymbol{Y}}^T - \boldsymbol{R}\overline{\boldsymbol{X}}^T \tag{12}$$

However, even with camera calibration, the issue of image distortion is still present, leading to a poor estimation of the marker location in the camera coordinate frame. Therefore, rigid transformation solely may be insufficient to yield an accurate estimation of the world coordinate values in the testing stage. Thus, other data-driven modeling methods that can incorporate additional nonlinearity into the mapping relationship $\boldsymbol{F}$ are also examined in this work.

2.2.2. Polynomial regression

Polynomial regression is a widely used approach to build the mapping relationship $\boldsymbol{F}$ between inputs and output responses, in which polynomial terms of inputs are used as regressors. In this work, X_c, Y_c, and Z_c are inputs, and X_w, Y_w, and Z_w are outputs, and $\boldsymbol{F}$ can be determined by

$$\mathbf{Y} = \boldsymbol{\xi}\boldsymbol{F} + \boldsymbol{v} \tag{13}$$

where $\boldsymbol{\xi} \in \mathfrak{R}^{n\times n_p}$ is a matrix composed of n_p regressors at the n observations. In this work, our regressors include the constant, linear (i.e., X_c, Y_c, and Z_c), and the second-order nonlinear terms

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

($X_cY_c, X_cZ_c, Y_cZ_c, {X_c}^2, {Y_c}^2$ and ${Z_c}^2$) of the input variables, and hence $n_p = 10$. $\boldsymbol{F} \in \mathfrak{R}^{n_p \times 3}$ is a matrix of model coefficients to be estimated, and $\boldsymbol{v} \in \mathfrak{R}^{n \times 3}$ is the matrix of the measurement errors for all the three outputs. The ordinary least-squares solution [26] to $\boldsymbol{F}$ is the best linear unbiased estimator (BLUE) and can be obtained by minimizing a cost function summing over all the squared residuals at each data point, yielding

$$\hat{\boldsymbol{F}} = (\boldsymbol{\xi}^T \boldsymbol{\xi})^{-1} \boldsymbol{\xi}^T \mathbf{Y} \tag{14}$$

Once $\hat{\boldsymbol{F}}$ is estimated in the offline training stage, it can be used for online real-time estimation of the marker center in the world coordinate frame, that is, X_{w0}, Y_{w0}, and Z_{w0}.

2.2.3. Artificial neural network model

The data-driven model in regression analysis can also be obtained by the machine learning approach [27], and specifically, the artificial neural network model (ANN) is adopted in this work. ANN consists of many neurons arranged in layers operating in parallel and can be mathematically described by a nonlinear weighted sum. The weights defining the strength of the connection between the neurons are trained through backpropagation to approximate the underlying mapping between the inputs and outputs. In this work, the multilayer feed-forward neural network (MLFNN) architecture is employed to construct three ANN models, respectively, for X_w, Y_w, and Z_w, that is, each individual output will be modeled separately. Thus, the model shown in Fig. 4 includes three inputs, three neurons in the hidden layer, and one output, and the number of neurons is determined through a trial-and-error process. In addition, the activation function of the hidden layer is the Tan-Sigmoid function. The activation function of the output layer is a linear function. Likewise, once the ANN model is trained, it can be used for online estimation of the marker center in the world coordinate frame.

2.2.4. Kriging interpolation model

Kriging, proposed by Krige and Sacks, is a data-driven interpolation technique to predict the response surface [28]. It was originally used in geostatistics and gradually applied to various engineering fields. In this work, the Kriging interpolation model is first developed to capture the mapping relationship $\boldsymbol{F}$ between the camera coordinate value, $\boldsymbol{x} = (X_c, Y_c,$ and $Z_c)$ and the world coordinate value $\boldsymbol{y} = (X_w, Y_w,$ or $Z_w)$ as presented above. The model has two components: a regression model $f(\boldsymbol{x})$ to estimate the global trend of the data landscape and a Gaussian process model $Z(\boldsymbol{x})$ with zero mean and variance σ^2 to capture the difference between the trend function and the true response surface. Therefore, the Kriging model reads

$$\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x}) = f(\boldsymbol{x}) + Z(\boldsymbol{x}) \tag{15}$$

The regression model $f(\boldsymbol{x})$ can be a known or unknown constant or a multivariate polynomial, yielding various categories of the Kriging model. The correlation matrix $\boldsymbol{\Psi}$ for observation data for the Gaussian process model is defined as

$$\boldsymbol{\Psi} = \begin{pmatrix} \psi(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}) & \cdots & \psi(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(n)}) \\ \vdots & \ddots & \vdots \\ \psi(\boldsymbol{x}^{(n)}, \boldsymbol{x}^{(1)}) & \cdots & \psi(\boldsymbol{x}^{(n)}, \boldsymbol{x}^{(n)}) \end{pmatrix} \tag{16}$$

where ψ is the correlation function for observation data, and the most widely used correlation functions include the Gaussian, exponential, spline, linear, and spherical functions. Furthermore, the hyperparameters (including σ^2 above) in the Kriging model are computed through the maximum likelihood estimation (MLE) [29]. The predicted mean of Kriging interpolation is given by

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{\alpha} + r(\boldsymbol{x})\Psi^{-1}(\boldsymbol{Y} - \boldsymbol{\xi}\boldsymbol{\alpha}) \tag{17}$$

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

where

$$\boldsymbol{M} = (b_1(x)\ b_2(x) \cdots b_{n_p}(x)) \tag{18}$$

$$\boldsymbol{\alpha} = (\boldsymbol{\xi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\xi})^{-1} \boldsymbol{\xi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{Y} \tag{19}$$

$$r(\boldsymbol{x}) = [\ \psi(\boldsymbol{x}, \boldsymbol{x}^{(1)}) \cdots \psi(\boldsymbol{x}, \boldsymbol{x}^{(n)})\ ] \tag{20}$$

and b_j is the jth polynomial regressor and α_j is the corresponding coefficient, and $0 \le j \le n_p$. $\boldsymbol{\xi}$ is the observation matrix as described above, and the estimated mean-squared error by the predictor is

$$s^2(\boldsymbol{x}) = \sigma^2 \left(1 - r\,(\boldsymbol{x})\ \boldsymbol{\Psi}^{-1} r\,(\boldsymbol{x})^T + \frac{1 - \boldsymbol{\xi}\boldsymbol{\Psi}^{-1} r\,(\boldsymbol{x})^T}{\boldsymbol{\xi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\xi}} \right) \tag{21}$$

In this work, the DACE (design and analysis of computer experiments) toolbox in MATLAB [30] is adopted for constructing the kriging model, and the linear regression model and Gaussian correlation model are used. Three Kriging models are developed for each component in the world coordinate values (X_w, Y_w, Z_w).

2.2.5. Kriging regression model

The predictor in Eq. (21) is for Kriging interpolation, well-suited for modeling more deterministic data. In contrast, the Kriging model can also be modified for regression to model data with significant noises and uncertainties that are governed by another Gaussian process $\eta(\boldsymbol{x})$ with the zero mean and covariance matrix $\boldsymbol{\Sigma}$

$$\eta \sim GP(0, \boldsymbol{\Sigma}) \tag{22}$$

Following a similar procedure above, the corresponding Kriging regression predictor is [26]

$$\hat{\boldsymbol{y}}\,(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{\alpha} + r\,(\boldsymbol{x}) \left(\boldsymbol{\Psi} + \frac{1}{\sigma^2} \boldsymbol{\Sigma} \right)^{-1} (\boldsymbol{Y} - \boldsymbol{\xi}\boldsymbol{\alpha}) \tag{23}$$

The type and distribution of noises produce different covariance matrices. The simplest form of the covariance matrix describing the noise and uncertainty is

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathrm{var}(y^{(1)}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathrm{var}(y^{(n)}) \end{pmatrix} \tag{24}$$

where $var(\boldsymbol{y}^{(i)})$ is the variance of the ith data observation. Furthermore, the noise can be assumed homogeneously distributed across all the input observations, and therefore, the covariance matrix can be set as $\boldsymbol{\Sigma} = 10^{\epsilon} \boldsymbol{I}_n$, where ϵ is used to quantify the amount of noise and can be determined as a hyperparameter using the MLE above. In this study, the ooDACE Toolbox is utilized to construct the model [31]. Likewise, three kriging regression models are constructed for predicting the world coordinate X_w, Y_w, and Z_w, respectively.

2.2.6. Hybrid models

In this work, hybrid models that combine rigid transformation and data-driven models are also developed to enhance positioning accuracy. Specifically, rigid transformation is first used to obtain intermediate values ($\tilde{X}_w$, $\tilde{Y}_w$, $\tilde{Z}_w$), which are then entered as the input to one of the data-driven models above, that is, polynomial regression, ANN, Kriging interpolation, and Kriging regression to further improve the world coordinate estimation (X_w, Y_w, Y_w).

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

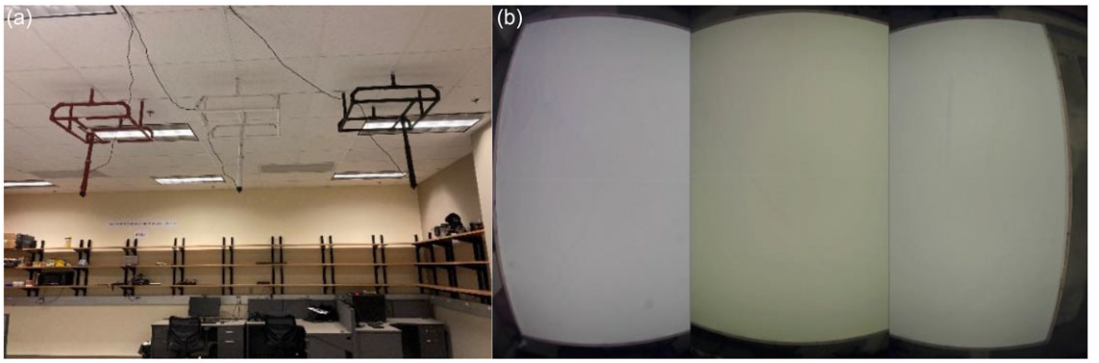

Figure 5. *(a) Camera installation on the custom mount and (b) images of floorboard taken by the three cameras.*

3. System hardware and experimental setup

As discussed above, the goal of this research is to develop a cost-effective indoor positioning system for robotics research and education. Figure 5(a) shows the three low-cost cameras installed on a custom mount attached to the ceiling for detecting and localizing the marker. The cameras are ELP 180-degree Fisheye cameras [32] with a unit price of around $50. Their resolution is set as 1920 × 1080. A low-end PC is also needed for image data acquisition and processing. Figure 5(b) presents the images of different areas of a white floorboard taken by these cameras, and the border regions between two adjacent cameras have partial overlap. To ensure the entire ArUco marker is within at least one camera view at any time, the overlap area needs to be larger than the ArUco marker. In other words, at any location of the floorboard, the camera coordinates (X_c, Y_c, Z_c) of the ArUco marker can be obtained using the image processing procedure in Eqs. (3), (4), and (5). The custom mounts are made of PVC pipes, on which the cameras can be moved along three perpendicular dimensions to adjust their orientations relative to the floorboard. However, it should be noted that the custom mount of the 3-axis motion used in our robotics research is not mandatory for the proposed indoor positioning system. In addition, the three mounts are painted in three different colors, red, white, and black, which also refer to the attached cameras hereafter.

During installation, all these cameras need to be aligned almost parallel to the floorboard. Therefore, a few markers (four used in this work) are placed at different locations to estimate the largest difference between their Z_c values within the view of a camera. Then the camera is adjusted manually to make the difference as small as possible. The adjustment process is repeated for all three cameras. However, eliminating the difference is almost impossible because of the image distortion, particularly with the low-cost cameras, and is also unnecessary as our data-driven models will correct and improve the estimation regardless. Therefore, when their difference is less than 5 cm, the camera is considered parallel with the floorboard.

To place the ArUco marker at different heights, that is, Z_w, relative to the floorboard, a small table, as shown in Fig. 6 is devised and 3D printed, which has a size of 20 cm (length) × 20 cm (width) × 5 cm (height). The marker used in this work is 20 cm (length) × 20 cm (width), and thus, it can be placed exactly on the top surface of the table. There is a small hole at the center of the bottom layer of the table, which is also the projected location of the marker's center when it is placed on the table. To continuously vary the marker's heights, a lab jack (a Scissor Stand Platform of 4" x 4") is used to lift the table up and down (Fig. 7). The height of the lab jack can be changed from 4.5 to 15 cm, and the exact height of the marker above the floorboard is measured by a ruler, which corresponds to the Z_w coordinate value in the world frame.

The floorboard in our experiment is made of 8 wooden boards (48 inches × 96 inches each) and can be entirely covered by the combined views of the three cameras (Fig. 5(b)). Thus, its total area is 192 inches × 192 inches (about 4.88 m × 4.88 m), as shown in Fig. 8. In this work, the coordinates of any

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

Figure 6. *The 3D printed table for holding the ArUco marker.*

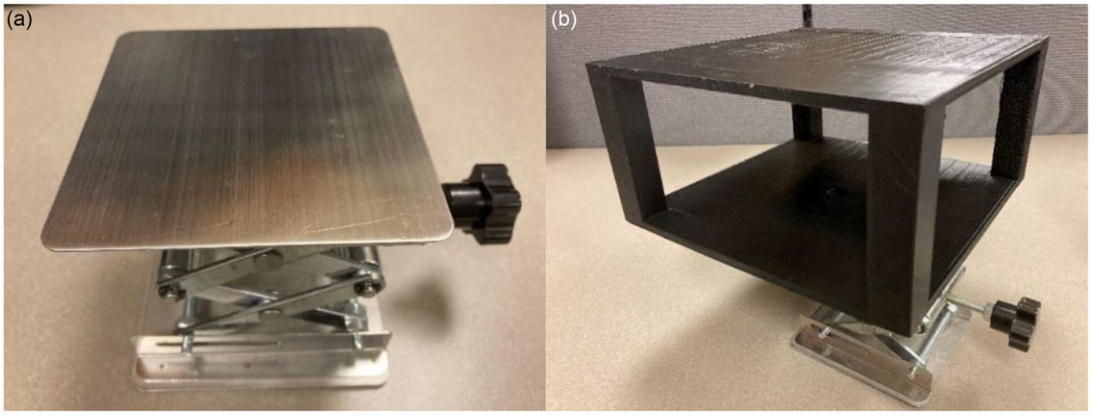

Figure 7. *(a) The lab jack to vary the heights of the ArUco marker and (b) the table on the lab jack.*

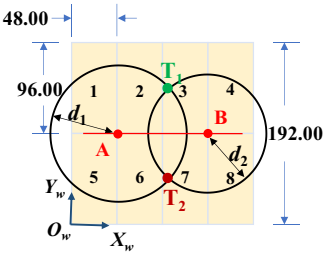

Figure 8. *Floorboard and indirect method to determine the location of the ArUco marker.*

location on the floorboard are measured indirectly using referenced points rather than directly from the origin. It is difficult to measure X_w and Y_w coordinate values directly from the origin because it requires X_w and Y_w axes to be set exactly perpendicular to each other. Further, when measuring the distance from a location to X_w and Y_w axes, the ruler also needs to be perpendicular to both axes. Both of these are not easy to achieve manually. Thus, a different approach that makes use of two referenced points is developed to determine the X_w and Y_w coordinates of the Aruco marker under the camera's view. The two reference points used in this work include point A (1.2192, 2.4384, 0) and point B (3.6576, 2.4384, 0) with a unit of meter. There are two reasons for choosing these two reference points. First, they are the intersection points of the wooden boards and can be easily found. There are kind of in the middle of the view area, and all locations on the floorboard are not far away from them.

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

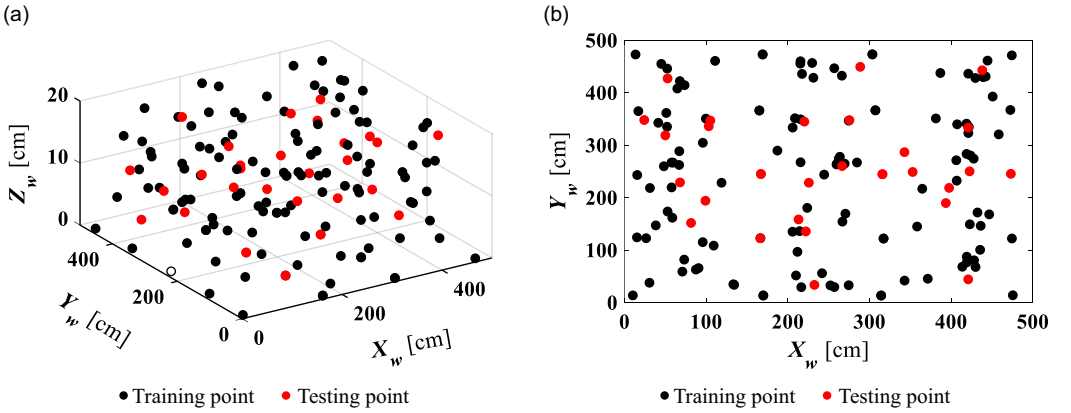

Figure 9. *3D and 2D views for training points and testing points (black and red points are used for training and testing, respectively).*

Specifically, we measure the distances, d_1 and d_2 from the marker location to Point A and Point B, which are given by

$$\begin{cases} (X_{w,g} - X_{w,A})^2 + (Y_{w,g} - Y_{w,A})^2 = d_1^2 \\ (X_{w,g} - X_{w,B})^2 + (Y_{w,g} - Y_{w,B})^2 = d_2^2 \end{cases} \tag{25}$$

where $X_{w,g}$ and $Y_{w,g}$ denote the world coordinates of the marker and will be determined by solving Eq. (25). The $Z_{w,g}$ coordinate value of the marker is known since the height of the table on which the marker is placed can also be measured. Once obtained, $(X_{w,g}, Y_{w,g}, Z_{w,g})$ will be used as the true value of the world coordinate for model training (Supplementary File; available online only). However, as shown in Fig. 8, the marker can be located at T_1 or T_2, and both of their distances to Point A and Point B satisfy Eq. (25). Thus, when we measure the distances, the position of the marker relative to the red line AB is also recorded. If the marker's location is above the red line AB, then $X_{w,g}$ and $Y_{w,g}$ coordinate values corresponding to T_1 are accepted. Otherwise, those of T_2 will be taken.

To train and identify the model $\boldsymbol{F}$ between (X_c, Y_c, Z_c) and (X_w, Y_w, Z_w), as presented in Section 2, a dataset containing true values of the world coordinates of the marker needs to be attained first for each camera. Within each camera's view, the marker is placed at 50 different locations. Then, its camera coordinate values are determined using Eqs (3), (4), and (5) by processing the collected images, and those in the world coordinate are manually measured following the procedure described above. Thus, 150 data pairs of true values of (X_w, Y_w, Z_w) and (X_c, Y_c, Z_c) are acquired. Out of the 50 pairs of data for each camera, 40 are randomly selected for model training, and the rest 10 are for testing. The 3D and 2D views of all the 150 locations of the ArUco markers in the world coordinate system are shown in Fig. 9(a) and (b), respectively, and each dot represents one marker location. Dots in black and red, respectively, denote the locations for training and testing model $\boldsymbol{F}$.

4. Experimental results

In this section, models above for $\boldsymbol{F}$ are evaluated using the testing data that is not present in the training process. Two sets of experiments within the 3D and 2D domains are conducted. In the former, the marker is also placed at different heights, while in the latter, the marker is on the floorboard, that is, $Z_w = 0$.

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

4.1. 3D positioning

The model performances are first evaluated for 3D positioning and the accuracy metric is defined as [2, 33]

$$\varepsilon = \sqrt{(X_{w,g} - \hat{X}_w)^2 + (Y_{w,g} - \hat{Y}_w)^2 + (Z_{w,g} - \hat{Z}_w)^2} \tag{26}$$

where $(X_{w,g}, Y_{w,g}, Z_{w,g})$ are the ground truth value of the world coordinate of the ArUco marker and are manually measured using Eq. (25) and following the procedure above. $(\hat{X}_w, \hat{Y}_w, \hat{Z}_w)$ are those estimated by our models and algorithms.

Table I compares the performance of various models for all three cameras (red, white, and black, as shown in Fig. 5) in mean accuracy, lower bound, upper bound, and standard deviation. Differences in performance among the three cameras are caused by the discrepancy in camera installation and intrinsic parameters of cameras. In total, nine different models/methods are listed in Table I. Method 1 is the rigid transformation method and is denoted RT hereafter. The mean accuracy of RT for red, white, and black cameras is 1.430, 2.344, and 3.422 cm, respectively. Method 2 is the polynomial regression (PR) method, and its mean accuracy for all three cameras is around 1.5 cm. The lower bound and standard deviation of PR are both lower than 1 cm, and the upper bound is less than 4 cm. Compared to RT, PR shows excellent improvement in lower bound, upper bound, and standard deviation. In Method 3, that is, RT + PR hybrid model, RT is applied to estimate the world coordinates $(\tilde{X}_w, \tilde{Y}_w, \tilde{Z}_w)$ first, and then these intermediate values are used by PR as the input to estimate the world coordinates again to further improve the accuracy. The mean accuracy, lower bound, upper bound, and standard deviation of RT + PR are the same as that of PR. Method 4, that is, ANN, is the machine learning-based regression model to estimate the world coordinates. The mean accuracy of ANN ranges from 1.5 to 2.0 cm, which is slightly worse than those of PR. The lower bound of ANN is also lower than 1 cm, and its upper bound and standard deviation are slightly higher than those of PR. In Method 5, that is, RT + ANN, the ANN model uses the estimated world coordinates from RT as inputs to estimate the world coordinates again. It seems that this hybrid model does not contribute to significant improvement in the mean accuracy. The accuracy of RT + ANN is between 1.5 and 1.8 cm, and its lower bound is also less than 1 cm. However, its upper bound and standard deviation are slightly inferior to that of ANN only. In Method 6, the Kriging Interpolation (KI) model is used to estimate the world coordinates. Compared to that of PR, the mean accuracy of KI is superior for the red camera but much worse for the white and black cameras. The upper bound and standard deviation of KI are also much higher than that of PR. Method 7 is the hybrid model combining RT + KI. Its mean accuracy is almost the same as that of KI. The differences in the lower bound, upper bound, and standard deviation between RT + KI and KI are indeed minor. Method 8 uses Kriging regression (KR) to estimate the coordinate value of the marker in the world frame. In contrast to KI, KR achieves better mean positioning accuracy for the black camera. However, the upper bound and standard deviation for the red and black cameras of KR are worse than that of KI. In Method 9, RT + KR forms a hybrid model to further improve the world coordinate estimation using the intermediate results from RT. Basically, there is no notable difference in the mean accuracy between KR and RT + KR. Their lower bound, upper bound, and standard deviation are also almost identical.

Through the comparison of these data-driven modeling methods, several interesting observations can be made. The combination of RT and other methods, such as RT + PR, RT + ANN, RT + KI, and RT + KR, makes a negligible contribution to mean accuracy improvement when compared to the single PR, ANN, KI, and KR. The difference in the mean accuracy among these nine methods is minor for the red camera but is noticeable for white and black cameras. PR and RT + PR are clearly top performers and achieve a mean accuracy of around 1.5 cm for all cameras and greatly outperform the other methods. And the lower bound, upper bound, and standard deviation for PR and RT + PR are the same. Compared to existing research works using the ArUco marker, the present system achieves a value of around 1.5 cm, a significant improvement in 3D indoor positioning accuracy.

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

Table I. *Performance of 9 different methods in 3D indoor positioning.*

		Mean accuracy (unit: cm)			Lower bound (unit: cm)			Upper bound (unit: cm)			Standard deviation (unit: cm)		
No.	**Methods**	**Red**	**White**	**Black**	**Red**	**White**	**Black**	**Red**	**White**	**Black**	**Red**	**White**	**Black**
1	RT	1.430	2.344	3.422	0.486	0.951	1.955	3.247	8.390	6.915	0.836	2.252	1.388
2	PR	1.517	1.506	1.482	0.865	0.383	0.193	2.702	3.923	3.012	0.650	1.073	0.930
3	RT + PR	1.517	1.506	1.482	0.865	0.383	0.193	2.702	3.923	3.012	0.650	1.073	0.930
4	ANN	1.577	1.694	2.076	0.725	0.714	0.819	3.482	4.289	5.380	0.861	1.077	1.385
5	RT + ANN	1.562	1.693	1.754	0.533	0.906	0.521	2.411	2.998	3.649	0.606	0.761	0.917
6	KI	1.306	2.297	2.898	0.401	0.534	1.273	3.548	5.430	5.250	0.970	1.506	1.260
7	RT + KI	1.312	2.294	2.804	0.427	0.529	1.055	3.600	5.429	5.227	0.980	1.507	1.319
8	KR	1.402	2.310	2.432	0.423	0.349	0.333	2.577	8.685	8.929	0.561	2.604	2.470
9	RT + KR	1.400	2.336	2.363	0.463	0.118	0.305	2.589	9.058	9.013	0.559	2.766	2.499

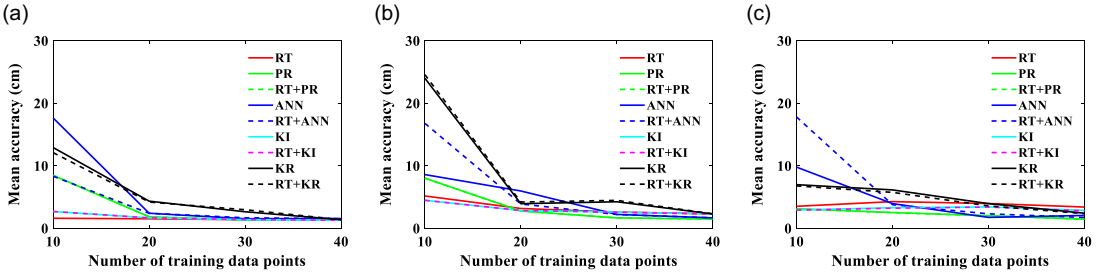

Figure 10. *The effect of the number of training data points on 3D positioning accuracy: (a) red, (b) white, and (c) black cameras.*

All these nine modeling methods in Table I perform well in positioning when the number of training data is 40 for each camera. However, the manual measurement and data collection are tedious and time-consuming and should be minimized subject to the positioning accuracy requirement. Therefore, the tradeoff between positioning accuracy and the training data size is also studied. As shown in Fig. 10, an analysis is conducted to investigate the influence of the number of training data points (10, 20, 30, and 40) on positioning accuracy. Figure 10(a), (b), and (c) show the mean accuracy for the red, white, and black cameras, respectively. Again, the training data points are selected randomly from the data pool in Fig. 9, and the locations of testing points remain the same. It is interesting to observe that the mean accuracy of these methods, in general, improves as more training data points are incorporated. All of them yield a mean accuracy of less than 10 cm when the number of training data points is 20 or more. In addition, even with the same method, the accuracy of different cameras can be different because the locations of selected training and testing data and the camera installation all affect model performance. The mean accuracy of ANN, RT + ANN, KR, and RT + KR deteriorates appreciably when training data are limited, while that of RT, PR, RT + PR, KI, and RT + KI maintain relatively constant with different amounts of training data. Especially, RT performs very consistently and even exceeds PR when only 10 data points are used, and its mean accuracy for all three cameras mostly keeps below 5 cm, which may be ascribed to physics/kinetics contained in its mathematical form in Eq. (6) that alleviates the demand for data. Our observation implies that when training data is scarce, and it is difficult to collect more, RT and KI may be more appropriate for modeling the mapping relationship $\boldsymbol{F}$.

4.2. 2D positioning

When the system is used for a mobile robot, and the height of the attached ArUco marker does not change appreciably, the estimation of the Z_w coordinate value is not necessary. Thus, the 2D positioning accuracy is more important, and its evaluation metric will be rewritten by removing the term associated with Z_w in Eq. (26)

$$\varepsilon = \sqrt{(X_{w,g} - \hat{X}_w)^2 + (Y_{w,g} - \hat{Y}_w)^2} \tag{27}$$

In Table II, the mean accuracy, lower bound, upper bound, and standard deviation for the nine methods in the 2D domain are listed. The mean accuracy of RT varies between 0.9 and 1.8 cm. The lower bound of RT is relatively small and only 0.4 cm. However, the upper bound is large and reaches 7.126 cm. Besides, its standard deviation is also larger than most methods, as shown in Table II. Among all methods, PR and RT + PR again perform the best and achieve the 2D mean accuracy of about 1 cm, and their lower bound, upper bound, and standard deviation are also smaller than those of the other methods. The mean accuracy of ANN, RT + ANN, KI, and RT + KI ranges from 1.1 to 1.8 cm, which is somewhat worse than that of PR and RT + PR. Correspondingly, their lower bound, upper bound, and standard deviation

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

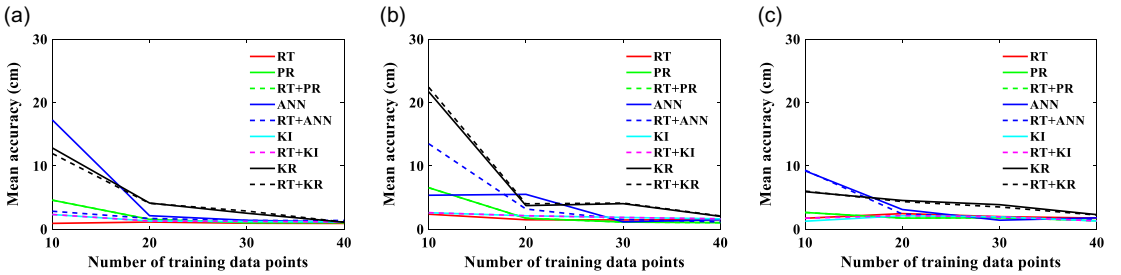

Figure 11. *The effect of the number of training data points on 2D positioning accuracy: (a) red, (b) white, and (c) black cameras.*

are slightly worse relative to PR and RT + PR. The mean accuracy of KR and RT + KR is the worst compared to other methods, and the discrepancy between the estimation and ground truth can reach 2.307 cm. Although their lower bound is lower, the upper bound and the standard deviation are high and, respectively, over 8 and 2.5 cm. Again, PR and RT + PR exhibit excellent positioning performance, and the combination of RT and PR does not offer apparent advantages over the PR only.

The effect of the number of training data on the mean accuracy of 2D positioning for red, white, and black cameras is shown in Fig. 11(a), (b), and (c), and the general trend and dependence on the training data size is similar to that of 3D positioning. The methods of ANN, RT + ANN, RK, and RT + RK perform poorly when the number of training data points is only 10. All these methods perform very well with a mean accuracy smaller than 10 cm when 20 or more data points are used for model training. RT, RT + PR, PR, KI, and RT + KI are able to keep relatively consistent positioning performance even with only ten training data points, while RT is particularly appealing. Similarly, when only a limited amount of data is available for model training, RT, KI, or RT + KI are preferred.

4.3. Verification with mobile robot experiment

Next, a test is carried out to verify that the indoor positioning system can be used to track moving robots in real time for research uses. The indoor positioning system proposed in this study is implemented on a laptop with an Intel(R) Core (TM) i5-7200U CPU @ 2.50 GHz. The mobile robot moves at a maximum speed of 0.26 m/s. The system processes three frames simultaneously from the three cameras within approximately 0.05 s, which allows real-time positioning and control of the robot. The ArUco marker is placed on the top of a mobile robot, and the centers of both are aligned. As shown in Fig. 12, "+" labels are placed at four different locations, which are, respectively, $P_1(0.5, 0.5)$, $P_2(0.5, 2.5)$, $P_3(2.5, 2.5)$, and $P_4(4.5, 4.5)$, all with a unit of meter. They are set as the waypoints for the mobile robot, and the starting position of the robot is around $P_1(0.5, 0.5)$, as shown in Fig. 12(a). In addition, the estimated location of the marker's center in the world coordinate frame, that is, $\hat{X}_{w0}$ and $\hat{Y}_{w0}$ is shown at the top-left corner of each subfigure in Fig. 12. Then the onboard PID controller uses the position information of $\hat{X}_{w0}$ and $\hat{Y}_{w0}$ to command the robot to go through the four waypoints above sequentially. In Fig. 12(b), it is clearly seen that the mobile robot moves from $P_1(0.5, 0.5)$ to $P_2(0.5, 2.5)$. In Fig. 12(c) and (d), the robot reaches $P_2(0.5, 2.5)$ and $P_3(2.5, 2.5)$ in tandem. Subsequently, the robot is on its way toward $P_4(4.5, 4.5)$, as shown in Fig. 12(e). Finally, the robot arrives at destination $P_4(4.5, 4.5)$ in Fig. 12(f). It should be noted that because of the partially overlapped view between two adjacent cameras, P_2 is present in both the white (middle) and black (right) cameras, and sometimes the robot appears simultaneously in two camera images (e.g., Fig. 12(b), (c), and (e)). This experiment verifies that the present indoor positioning system can track the mobile ground robot and provides accurate position information in real time for robot control.

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

Table II. *Performance of 9 different methods in 2D indoor positioning.*

		Mean accuracy (unit: cm)			Lower bound (unit: cm)			Upper bound (unit: cm)			Standard deviation (unit: cm)		
No.	**Methods**	**Red**	**White**	**Black**	**Red**	**White**	**Black**	**Red**	**White**	**Black**	**Red**	**White**	**Black**
1	RT	0.923	1.540	1.769	0.125	0.393	0.234	1.945	7.126	6.492	0.626	2.028	2.036
2	PR	1.044	1.035	1.354	0.321	0.280	0.150	1.783	2.074	3.004	0.452	0.569	0.966
3	RT + PR	1.044	1.035	1.354	0.321	0.280	0.150	1.783	2.074	3.004	0.452	0.569	0.966
4	ANN	1.144	1.505	1.798	0.407	0.625	0.421	2.681	4.228	5.322	0.673	1.077	1.462
5	RT + ANN	1.389	1.162	1.566	0.531	0.410	0.335	2.300	2.989	3.521	0.601	0.716	1.017
6	KI	1.161	1.649	1.312	0.255	0.432	0.349	2.764	5.429	2.775	0.800	1.501	0.895
7	RT + KI	1.159	1.649	1.314	0.253	0.434	0.354	2.766	5.428	2.802	0.800	1.500	0.903
8	RK	1.115	2.037	2.307	0.322	0.125	0.139	1.833	8.554	8.927	0.517	2.671	2.527
9	RT + RK	1.126	2.101	2.242	0.347	0.117	0.082	1.826	8.791	9.011	0.506	2.738	2.558

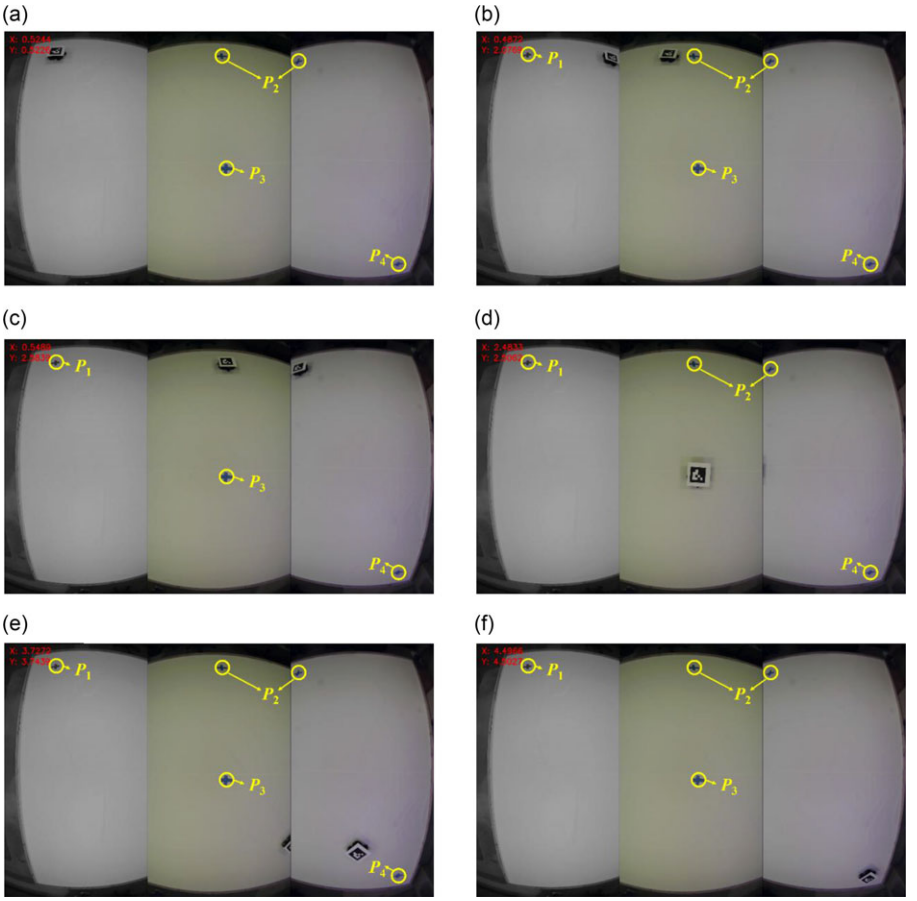

Figure 12. *Experiment to verify the feasibility of using our indoor positioning system for robot control and research uses.*

5. Conclusion

In this paper, a low-cost and accurate positioning system, along with a novel pipeline to combine image processing and data-driven modeling, is proposed for robotics research and education. Both hardware and algorithms, in conjunction with software–hardware interfacing, are developed for automated image acquisition and processing, computing, and positioning, all in real time. The key contribution of the present effort is to push the limit of positioning accuracy through optimally selected algorithms while minimizing the overall cost of hardware and installation, which makes it more affordable to the broad robotics community.

Our system includes multiple overhead cameras to acquire images of the ArUco marker. OpenCV is employed to extract its normalized image coordinates. A new numerical procedure is formulated to convert them to the camera coordinate frame. The mapping of the marker's position from the camera coordinate to the world coordinate is tackled by data-driven models. Various modeling techniques are interrogated, including rigid transformation, polynomial regression, artificial neural network, Kriging interpolation and regression, and hybrid models. A dataset is also constructed, which contains manually measured data pairs at 150 locations. The trained model enables real-time estimation of the world coordinate values of the ArUco marker (and its robot carrier).

Experimental studies are also carried out for both 3D and 2D positioning. Polynomial regression, although straightforward to implement, exceeds most other methods and yields a positioning accuracy

https://doi.org/10.1017/S0263574723000589 Published online by Cambridge University Press

of about 1.5 cm. Rigid transformation and Kriging interpolation preserve consistent performance even if only ten train data points are used, yielding a mean accuracy mostly within 5 cm. Another test of the mobile robot is also performed, which uses the real-time position information provided by our system for navigation and control. The present system is shared as an open-source tool in the public domain, and it is anticipated to contribute to robotics studies and education in resource-limited environments, particularly those in geographically underdeveloped regions.

Author contributions. Conceptualization: Junlin Ou, Seong Hyeon Hong, Yi Wang; methodology: Junlin Ou; formal analysis and investigation: Junlin Ou, Seong Hyeon Hong, Tristan Kyzer, Haizhou Yang, Yi Wang; writing – original draft preparation: Junlin Ou; writing – review and editing: Seong Hyeon Hong, Haizhou Yang, Xianlian Zhou, Yi Wang.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Conflicts of interest. The authors declare no conflicts of interest exist.

Data availability statement. The data and material used for this study is available in Github.

Ethical approval. Not applicable.

Supplementary material. To view supplementary material for this article, please visit https://doi.org/10.1017/S0263574723000589.

References

[1] A. R. Jiménez and F. Seco, "Comparing Decawave and Bespoon UWB Location Systems: Indoor/Outdoor Performance Analysis," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016 Oct 4* (IEEE, 2016) pp. 1–8.
[2] R. Amsters, E. Demeester, N. Stevens, Q. Lauwers and P. Slaets, "Evaluation of Low-Cost/High-Accuracy Indoor Positioning Systems," *Proceedings of the 2019 International Conference on Advances in Sensors, Actuators, Metering and Sensing (ALLSENSORS), Athens, Greece 2019 Feb 24* (2019) pp. 24–28.
[3] J. Kuang, X. Niu, P. Zhang and X. Chen, "Indoor positioning based on pedestrian dead reckoning and magnetic field matching for smartphones," *Sensors* **18**(12), 4142 (2018).
[4] H. Ju, S. Y. Park and C. G. Park, "A smartphone-based pedestrian dead reckoning system with multiple virtual tracking for indoor navigation," *IEEE Sens. J.* **18**(16), 6756–6764 (2018).
[5] R. Ali, R. Liu, A. Nayyar, B. Qureshi and Z. Cao, "Tightly coupling fusion of UWB ranging and IMU pedestrian dead reckoning for indoor localization," *IEEE Access* **9**, 164206–164222 (2021).
[6] M. Mareedu, S. Kothacheruvu and S. Raghunath, "Indoor Navigation Using Ultra-Wideband Indoor Positioning System (IPS)," AIP Conference Proceedings, 2021 Dec 1, *AIP Publishing LLC*, vol. 2407 (2021) pp. 020022.
[7] Z. Zhang, M. Lee and S. Choi, "Deep-learning-based wi-fi indoor positioning system using continuous CSI of trajectories," *Sensors* **21**(17), 5776 (2021).
[8] P. Bencak, D. Hercog and T. Lerher, "Indoor positioning system based on bluetooth low energy technology and a nature-inspired optimization algorithm," *Electronics* **11**(3), 308 (2022).
[9] T. Zhou, J. Ku, B. Lian and Y. Zhang, "Indoor positioning algorithm based on improved convolutional neural network," *Neural Comput. Appl.* **34**(9), 6787–6798 (2022).
[10] S. Yan, Y. Su, A. Sun, Y. Ji, J. Xiao and X. Chen, "Low-Cost and Lightweight Indoor Positioning Based on Computer Vision," *2022 4th Asia Pacific Information Technology Conference 2022 Jan 14* (2022) pp. 169–175.
[11] Z. Y. Ng, Indoor-Positioning for Warehouse Mobile Robots Using Computer Vision (Doctoral dissertation, UTAR).
[12] J. Kunhoth, A. Karkar, S. Al-Maadeed and A. Al-Ali, "Indoor positioning and wayfinding systems: A survey," *Hum.-Centric Comput. Infor. Sci.* **10**(1), 1–41 (2020).
[13] J. Kuang, X. Niu and X. Chen, "Robust pedestrian dead reckoning based on MEMS-IMU for smartphones," *Sensors* **18**(5), 1391 (2018).
[14] B. Wang, X. Liu, B. Yu, R. Jia and X. Gan, "Pedestrian dead reckoning based on motion mode recognition using a smartphone," *Sensors* **18**(6), 1811 (2018).
[15] X. Liu, Z. Jiao, L. Chen, Y. Pan, X. Lu and Y. Ruan, "An enhanced pedestrian dead reckoning aided with DTMB signals," *IEEE Trans. Broadcast* **68**(2), 407–413 (2022).
[16] A. R. Ruiz and F. S. Granja, "Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis," *IEEE Trans. Inst. Meas.* **66**(8), 2106–2117 (2017).
[17] P. Sthapit, H. S. Gang and J. Y. Pyun, "Bluetooth Based Indoor Positioning Using Machine Learning Algorithms," *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2018 Jun 24* (IEEE, 2018) pp. 206–212.

[18] Z. Han, Z. Wang, H. Huang, L. Zhao and C. Su, "WiFi-based indoor positioning and communication: Empirical model and theoretical analysis," *Wireless Commun. Mobile Comput.,* **2022** 1–12 (2022).
[19] S. Jia, L. Ma, S. Yang and D. Qin, "A novel visual indoor positioning method with efficient image deblurring," *IEEE Trans. Mobile Comput.*, 1–1 (2022).
[20] X. Liu, H. Huang and B. Hu, "Indoor visual positioning method based on image features," *Sens. Mater.* **34**(1), 337–348 (2022).
[21] Y. Li, Z. Ghassemlooy, X. Tang, B. Lin and Y. Zhang, "A VLC smartphone camera based indoor positioning system," *IEEE Photonic Tech. Lett.* **30**(13), 1171–1174 (2018).
[22] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead and N. Vitzilaios, "Fiducial markers for pose estimation," *J. Intell. Rob. Syst.* **101**(4), 1–26 (2021).
[23] G. C. La Delfa, S. Monteleone, V. Catania, J. F. De Paz and J. Bajo, "Performance analysis of visualmarkers for indoor navigation systems," *Front. Inform. Technol. Electron. Eng.* **17**(8), 730–740 (2016).
[24] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal.* **Sep**(5), 698–700 (1987).
[25] A. Kurobe, Y. Sekikawa, K. Ishikawa and S. H. Corsnet, "3d point cloud registration by deep neural network," *IEEE Robot. Autom. Lett.* **5**(3), 3960–3966 (2020).
[26] A. Sobester, A. Forrester and A. Keane, *Engineering Design Via Surrogate Modelling: A Practical Guide* (John Wiley & Sons, 2008).
[27] N. Kato, B. Mao, F. Tang, Y. Kawamoto and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Commun.* **27**(3), 96–103 (2020).
[28] H. Yang, S. H. Hong, G. Wang and Y. Wang, "Multi-fidelity reduced-order model for GPU-enabled microfluidic concentration gradient design," *Eng. Comput.* 1–19 (2022).
[29] C. R. Dietrich and M. R. Osborne, "Estimation of covariance parameters in kriging via restricted maximum likelihood," *Math. Geol.* **23**(1), 119–135 (1991).
[30] S. N. Lophaven, H. B. Nielsen and J. Søndergaard. *DACE: A Matlab Kriging Toolbox* (IMM, Informatics and Mathematical Modelling, The Technical University of Denmark, Lyngby, Denmark, 2002).
[31] I. Couckuyt, T. Dhaene and P. Demeester, "ooDACE toolbox: A flexible object-oriented Kriging implementation," *J. Mach. Learn. Res.* **15**, 3183–3186 (2014).
[32] T. Kyzer, Instrumentation and Experimentation Development for Robotic Systems (Doctoral dissertation, University of South Carolina).
[33] S. H. Hong, J. Ou and Y. Wang, "Physics-guided neural network and GPU-accelerated nonlinear model predictive control for quadcopter," *Neural Comput. Appl.* **13**(1), 1–21 (2022).

Cite this article: J. Ou, S. H. Hong, T. Kyzer, H. Yang, X. Zhou and Y. Wang (2023). "A low-cost indoor positioning system based on data-driven modeling for robotics research and education", Robotica **41**, 2648–2667. https://doi.org/10.1017/S0263574723000589