Robotica (2025), **43**, pp. 3080–3093
doi:10.1017/S0263574725102117

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

RGB-D image-based real-time pose estimation algorithm for mobile robots with rectangular body

Haoran Tang, Bo Wang, Xiaofei Zhou, Zhimin Han and Qiang Lv

School of Automation and International Joint Research Laboratory for Autonomous Robotic Systems, Hangzhou Dianzi University, Hangzhou, China
Corresponding author: Bo Wang; Email: wangbo@hdu.edu.cn

Received: 10 September 2024; **Revised:** 1 July 2025; **Accepted:** 14 July 2025; **First published online:** 11 August 2025

Keywords: pose estimation; object detection; mobile robot

Abstract

In this study, we introduce a real-time pose estimation for a class of mobile robots with rectangular body (e.g., the common automatic guided vehicles), by integrating odometry and RGB-D images. First, a lightweight object detection model is designed based on the visual information. Then, a pose estimation algorithm is proposed based on the depth value variations within the target region that exhibit specific patterns due to the robot's three-dimensional geometry and the observation perspective (termed as "differentiated depth information"). To improve the robustness of object detection and pose estimation, a Kalman filter is further constructed by incorporating odometry data. Finally, a series of simulations and experiments are conducted to demonstrate the method's effectiveness. Experiments show that the proposed algorithm can achieve a speed over 20 Frames Per Second (FPS) together with a good estimation accuracy on a mobile robot equipped with an Nvidia Jetson Nano Developer KIT.

1. Introduction

Autonomous mobile robots, such as automatic guided vehicles, are receiving much attention from researchers due to their significant enhancement of task execution efficiency in applications within the military [1], agriculture [2], and other fields [3–7], of which a critical technology is the pose estimation technology. Pose estimation aims to provide precise front-end information for the subsequent control and decision task of robots by accurately estimating the pose of itself or the object, where some typical applications include precise navigation in autonomous driving [8, 9] and the effective object grasping for robotic arms [10, 11]. Pose estimation technology has always been a current research hotspot in the field of robotics.

Up to now, many algorithms that depend on various sensors (e.g., inertial sensors, GPS, and radar) have been proposed for pose estimation problems [12–15]. For instance, authors in ref. [12] propose an algorithm by using the data obtained from GPS. Reference [15] introduces a real-time pose estimation method based on the data obtained by multi-input multi-output radar. In addition to above-mentioned methods based on the single type of sensor, integrating multiple sensors (especially the vision sensor) in robots has become accessible with the rapid development of sensors with low prices. By fusing information from multiple sources, the pose estimation can be significantly improved. For example, ref. [16] proposes a monocular vision-inertial state estimator to achieve six-degree-of-freedom pose estimation based on data from a monocular camera and an IMU. Authors in ref. [13] propose a 3D motion pose estimation method based on data obtained from an IMU and a monocular camera, which utilize the IMU to enhance camera capabilities and improve pose estimation accuracy. When vision information is used in the pose estimation, the above-mentioned studies typically start by feature matching in grayscale images to gather the object information. Then, the pose is further estimated using multiple reference object poses by employing a database to compute image similarity [17, 18]. This process has some limitations.

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

For details, if the number of reference objects is too large, the matching process is time-consuming. Conversely, if there are too few matches, the estimation accuracy decreases.

Recently, some deep learning-based models have been proposed to directly estimate target poses from RGB images. For example, ref. [19] proposes the DSOAE-Net, which employs a multimodal R6D representation to estimate the pose and achieves good speed and accuracy across multiple datasets for uncooperative known space objects. Reference [20] develops an end-to-end architecture that regresses 3D coordinates from 2D detections to estimate 6D poses and achieves good precision and real-time performance on the LINEMOD [21] and OCCLUSION [22] datasets. These methods leverage GPU acceleration for the real-time estimation performance. However, they may lose the accuracy in the depth estimation if only RGB images are used, which is important to mobile robots. In order to improve the accuracy, the industry has chosen to divide the object pose estimation task into two stages: object detection and pose estimation. The author in ref. [23] compares synthetic data generation methods, which trains Mask R-convolutional neural network (CNN) for object detection with more realistic datasets and performs pose estimation with autoencoders based on depth differences. Reference [24] presents a method for instance-level object detection and 3D pose estimation by training a CNN with synthetic data. Though, its effectiveness has been verified through experiments, its difficult to deploy and perform inference of deep models on the GPU of an mobile embedded platform.

This study focuses on the pose estimation for a class of mobile robots with rectangular body. The pose estimation contains two steps: the image information-based object detection together with the odometry and depth information based pose estimation. Considering that the deep learning-based object detection method has become the current mainstream, this study will adopt such method to achieve the object detection on the mobile robot. For deep learning-based object detection, the leading real-time object detectors include the YOLO series [25–29] and cutting-edge models like boundary-aware remote sensing detectors [30] and the lightweight Transformer-based EF-DETR [31]. Despite these models' good real-time performance, their inference speed is still hard to meet the demand for embedded platforms with limited battery capacity and computing power [32]. As for the pose estimation, traditional methods usually estimate the object pose based on parallax formed by pixel transformations of reference objects in RGB images [33, 34]. However, under extreme low-light conditions, the reduced signal-to-noise ratio of image sensors leads to underexposure and loss of gradient details, which significantly degrade the accuracy of visual feature matching [35–37]. These physical limitations highlight the necessity of incorporating depth and odometry data to achieve more robust and accurate pose estimation.

In this study, we introduce a real-time pose estimation for a class of mobile robots with rectangular body by integrating odometry, vision, and depth data. Contributions are as follows:

1. We propose a lightweight vision-based object detection model suitable for embedded platforms with limited computing power. This model effectively reduces the computational burden and meets real-time requirements without compromising the accuracy of pose estimation.
2. Based on the obtained object detection model, a real-time pose estimation method for a class of mobile robots with rectangular body is further proposed. First, a pose estimation algorithm is proposed based on the differentiated depth information caused by the object's geometric features. Then, the odometry information is integrated into a Kalman filter algorithm to enhance the robustness and accuracy of the estimation.

To show the effectiveness of the proposed algorithm, we first constructed a simulation environment in Gazebo that closely mimics real-world scenarios and deployed our algorithm within a pose estimation system composed of two mobile robots to evaluate the estimation performance. Additionally, we conduct the experiments with two mobile robots in a real physical environment to further validate our approach. The experiment shows that the proposed algorithm can achieve a speed over 20 Frames Per Second (FPS) on a basic version Jetson Nano mobile GPU. Both simulation and experimental results show that the algorithm proposed in this study can achieve the object detection and pose estimation rapidly and accurately.

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

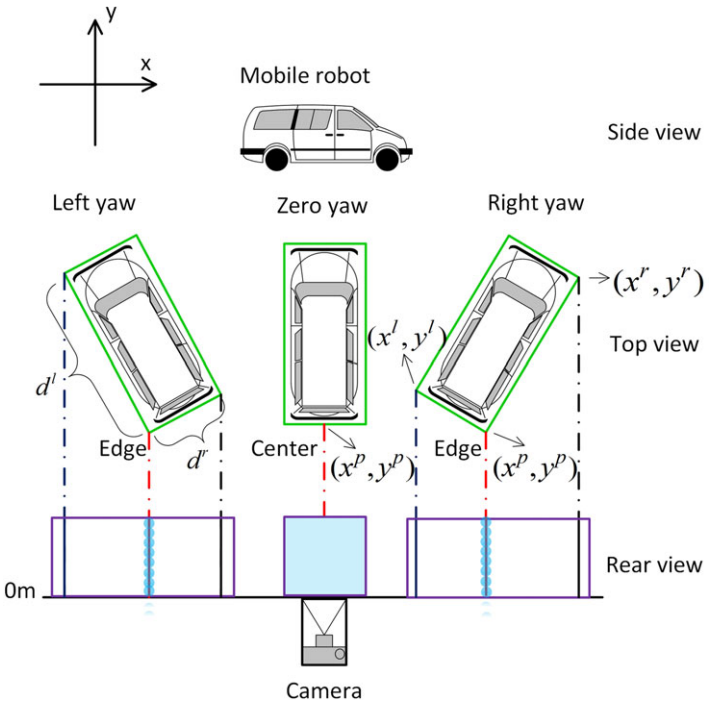


Figure 1. *Schematic diagram of the observation for the robot in the forward view of the observation platform.*

2. Problem formulation

We consider a mobile robot object with nonholonomic dynamics in the discrete-time form:

$$q_t = Aq_{t-1} + B_t u_t. \tag{1}$$

In (1), $q_t = [x_t, y_t, \theta_t]^T$ represents the pose of the robot in world (global) coordinate system with $x_t \in \mathbb{R}$, $y_t \in \mathbb{R}$ being the central position coordinates and $\theta_t \in [-\pi, \pi)$ being the orientation. $u_t = [v_t, \omega_t]^T$ is the control input with $v_t \in \mathbb{R}$ being the linear velocity and $\omega_t \in \mathbb{R}$ being the angular velocity. $A = I$ with I being the identity matrix and $B_t = \begin{bmatrix} \cos\theta_t & 0 \\ \sin\theta_t & 0 \\ 0 & 1 \end{bmatrix} dt$, respectively, represent the state transition matrix and control input matrix with dt representing the sampling time interval. We suppose the position x, y is located at the center of the robot.

In this study, we suppose the object has rectangular body shown in Figure 1. Suppose that the length denoted by r_l and the width denoted by r_w w.r.t. the robot are known. Besides, we suppose the orientation $\theta_t \in (0, \pi)$, where for the case $\theta_t \in [-\pi, 0]$, it can be easily judged through the object's movement direction. To achieve the pose estimation for such object, we suppose the observation platform can use some sensors (e.g., the RGB-D camera) to get the RGB-D images and velocity odometry information for the object. However, the measurement is usually not correct due to the noise effect and model accuracy. Besides, the real-time performance is very important for practical platform. Considering these points, the goal of this study is to design a real-time pose estimation method together with a good accuracy for the robot object.

Remark 1. *For the observation platform, its pose is usually known. As a result, if it can get the measurement under its own local coordinate system, the global measurement for the object can also be obtained. Without loss of generality, we directly suppose the observation platform has the world coordinate system.*

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

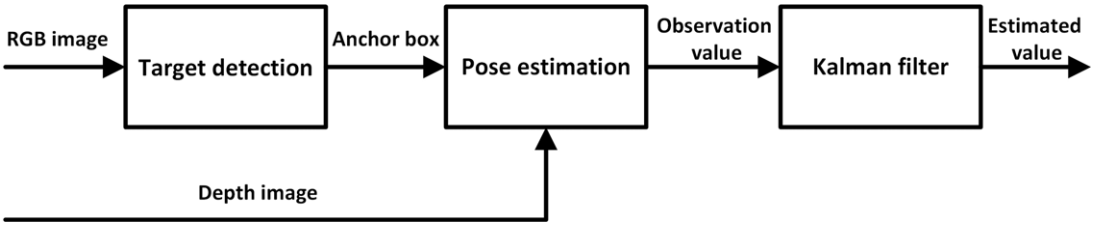


Figure 2. *Block diagram of the proposed algorithm.*

3. Algorithm design

In this section, we will give the complete pose estimation algorithm design.

Let (x^l, y^l), (x^r, y^r), and (x^p, y^p), respectively, represent the leftmost coordinate, the rightmost coordinate and the coordinate with the smallest depth on the robot's top view in the forward view of the observation platform's coordinate system (see Figure 1 for details). Obviously, if the orientation $\theta \neq \frac{\pi}{2}$ and $\theta \in (0, \pi)$, (x^p, y^p) is unique. Otherwise, (x^p, y^p) may not be unique. For this case, we let (x^p, y^p) represent the relative position from the robot's tail/head center to the observation platform. Let

$$d^l = \sqrt{(x^l - x^p)^2 + (y^l - y^p)^2},$$
$$d^r = \sqrt{(x^r - x^p)^2 + (y^r - y^p)^2}. \tag{2}$$

By (2), it can be concluded that if $\theta \neq \frac{\pi}{2}$ and $\theta \in (0, \pi)$, $r_l = \max(d_l, d_r)$, and $r_w = \min(d_l, d_r)$. Then, if the accurate coordinate information can be obtained, we can obtain the following conclusion.

Proposition 1. *Suppose that* $\theta_t \in (0, \pi)$ *and positions* (x^l, y^l), (x^r, y^r), (x^p, y^p) *can be measured accurately. Then, the following statements are true:*

1. *if* $d^l = d^r$, $(x, y, \theta) = \left(x^p, y^p + \frac{r_l}{2}, \frac{\pi}{2}\right)$;
2. *if* $d^l < d^r$, $(x, y, \theta) = \left(\frac{x^l + x^r}{2}, \frac{y^l + y^r}{2}, \arctan\left(\frac{y^r - y^p}{x^r - x^p}\right)\right)$;
3. *if* $d^l > d^r$, $(x, y, \theta) = \left(\frac{x^l + x^r}{2}, \frac{y^l + y^r}{2}, \pi + \arctan\left(\frac{y^r - y^p}{x^r - x^p}\right)\right)$.

Based on Figure 1, the proof can be easily obtained and is omitted here.

To estimate (x, y, θ) for the object, one should first achieve the object detection and then give the pose estimation, where the whole flow chart is illustrated in Figure 2. The first module is object detection. The second module involves pose estimation for objects based on the depth difference information caused by the object's shape. Finally, Kalman filter is employed to yield smoother and more robust pose estimation results. Detailed descriptions of each module are shown by the following subsections.

3.1. A lightweight object detection model

In order to design a lightweight single-object detection model that can run smoothly on embedded platforms, this study designs a one-stage deep learning based object detection model (see Figure 3 for details) inspired by YOLO series [25].

This model adopts the FPN network structure [38], which comprises backbone network, neck network, and detection head. The network uses low-level and high-level features to improve object detection accuracy across scales. For the feature extraction network (i.e., the backbone network), the CBR and CSP modules (shown in the top of Figure 3) are introduced to reduce computational burden while enhancing learning ability. The detailed improvements are shown as follows:

1. integrating CSP modules to reduce computational burden while enhancing learning ability.
2. gradually increase the number of convolutional channels to avoid instability during training.

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

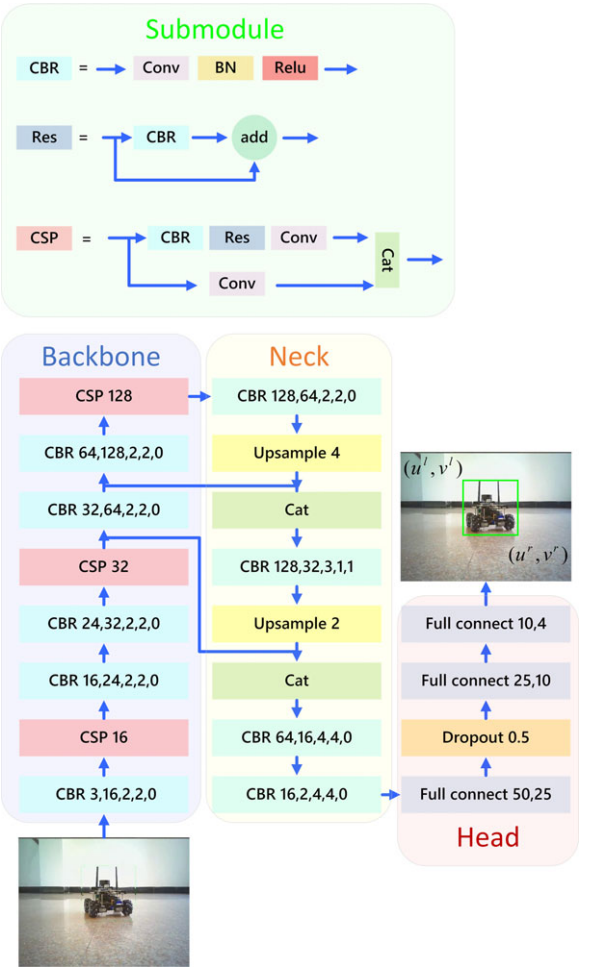


Figure 3. *A lightweight object detection model.*

3. downsampling with convolutional kernel size and stride of 2 to extract high-level semantic information.
4. using the feature fusion under top-down propagation and lateral connections to ensure rich semantic and precise positioning.

Unlike traditional object detection, this study focuses solely on single-object detection, where the evaluation relies solely on the *CIoU Loss* function.

3.2. Direct pose estimation based on RGB-D images

In this part, we will introduce a method to estimate the pose for the mobile robot object based on RGB-D images.

Let (u, v) represent the pixel coordinate in the image coordinate system with $f(u, v)$ being the corresponding depth. The transformation from image coordinates (u, v) to world coordinates (x, y, z) can be expressed as [39]

$$[x, y, z]^T = K^{-1} * [u, v, 1]^T * f(u, v), \tag{3}$$

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

where K represents the intrinsic matrix of the depth camera. Let (u^l, v^l) and (u^r, v^r), respectively, represent the upper left and lower right corners of the anchor box w.r.t. the object (see the output of the detection model in Figure 3 for details).

First, we will give the estimation for coordinates (x^p, y^p), (x^l, y^l), and (x^r, y^r). Let U^p represent a set consisting of all u-axis coordinates with the smallest depth, where each element can be obtained through the following formula:

$$u = \arg\min_{u \in [u^l, u^r]} f(u, v), \forall v \in [v^l, v^r]. \tag{4}$$

Let μ^p and δ^p, respectively, denote the sample mean and variance w.r.t. the elements in U^p, which are calculated as

$$\mu^p = \frac{1}{|U^p|} \sum_{u \in U^p} u, \quad \delta^p = \frac{1}{|U^p|} \sum_{u \in U^p} (u - \mu^p)^2. \tag{5}$$

Then, (x^p, y^p) can be estimated by (3) with $(u, v) = (\mu^p, \frac{v^l + v^r}{2})$. Let U^l represent a set of 3D points (x, y, z) corresponding to pixels in the depth image. These points are computed by applying the transformation in (3) to all pixel coordinates (u, v) that satisfy the following conditions: (1) $u \in [u^l, \mu^p]$, that is, along the horizontal axis from the leftmost edge of the detected bounding box to the intermediate reference point with minimum depth; (2) $v = \frac{v^l + v^r}{2}$, which corresponds to the vertical midpoint of the bounding box; and (3) the associated depth value $f(u, v)$ is valid (i.e., non-null). That is, U^l consists of all valid 3D points along the horizontal line at the center of the bounding box, from the left boundary to the point of minimal depth. Then, the point (x^l, y^l, z^l) can be estimated as the 3D point in U^l with the minimum x-coordinate:

$$(x^l, y^l, z^l) = \arg\min_{(x,y,z) \in U^l} x. \tag{6}$$

Let U^r represent a set w.r.t. (x, y), where each element is obtained by (3) with $u \in [\mu^p, u^r]$ and $v = \frac{v^l + v^r}{2}$ while satisfies that the obtained depth value z is no-null. Then, (x^r, y^r) can be estimated through the similar computation for (6) with

$$(x^r, y^r, z^r) = \arg\max_{(x,y,z) \in U^r} x. \tag{7}$$

Second, we will adopt the least square method to give the orientation estimation. Based on the estimated (x^p, y^p), (x^l, y^l) and (x^r, y^r), we can obtain the estimation of d^l and d^r by (2). Given the linear regression equation $f(x) = kx + b$, then, we can obtain two pairs of the estimation w.r.t. (k, b) by solving the following least square problem:

$$\begin{cases} (k^r, b^r) = \arg\min_{k,b} \sum_i (f(x^i) - y^i)^2, s.t.\ (x^i, y^i) \in U^r, \\ (k^l, b^l) = \arg\min_{k,b} \sum_i (f(x^i) - y^i)^2, s.t.\ (x^i, y^i) \in U^l. \end{cases} \tag{8}$$

The above optimization problems have closed-form solutions with

$$k = \frac{\sum_i (x^i - \bar{x})(y^i - \bar{y})}{\sum_i (x^i - \bar{x})^2}, \quad b = \bar{y} - k\bar{x}, \tag{9}$$

where $\bar{x}$ and $\bar{y}$ denote the sample means of the x^i and y^i values in S, respectively. Then, the orientation θ can be estimated as

$$\theta = \begin{cases} \arctan(k^r), & \text{if } d^l < d^r, \\ \pi + \arctan(k^l), & \text{otherwise.} \end{cases} \tag{10}$$

Finally, we can estimate the object coordinates as follows. When the orientation is orthogonal (i.e., $\theta = \frac{\pi}{2}$), the center is set to $(x^p, y^p + \frac{r_l}{2})$. For non-orthogonal cases, as stated in Proposition 1, the left and right boundaries are fitted as lines $y = k^l x + b^l$ and $y = k^r x + b^r$. The horizontal coordinate is

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

Algorithm 1. The implementation of the RGB-D image based pose estimation algorithm

Input: RGB-D image, depth camera intrinsic matrix K, pixel coordinates (u^l, v^l), and (u^r, v^r) obtained from the detection model.

Output: Object coordinates (x, y) and orientation θ

1. **Estimation for** (x^p, y^p)**,** (x^l, y^l) **and** (x^r, y^r)
   (a) Calculate the mean μ^p and variance δ^p of the set U^p, where U^p is obtained based on (4);
   (b) Estimate (x^p, y^p) by (3) with $(u, v) = (\mu^p, \frac{v^l+v^r}{2})$;
   (c) Calculate the set U^l by (3) with $u \in [u^l, \mu^p]$ and $v = \frac{v^l+v^r}{2}$;
   (d) Calculate the set U^r by (3) with $u \in [\mu^p, u^r]$ and $v = \frac{v^l+v^r}{2}$;
   (e) Estimate (x^l, y^l) and (x^r, y^r) by (6) and (7), respectively;
2. **Estimation for the orientation** θ
   (a) Estimate d^l and d^r based on (x^p, y^p), (x^l, y^l), and (x^r, y^r);
   (b) Solve the least squares problem to obtain (k^l, b^l) and (k^r, b^r) by (8);
   (c) Based on the obtained parameters (k^l, b^l), (k^r, b^r), d^l, and d^r to estimate the orientation θ by (10).
3. **Estimation for coordinates** (x, y)
   (a) Estimate coordinates (x, y) of the object by (11).

Algorithm 2. Kalman filter based pose estimation by fusing odometry data

Inputs: initial state estimation q_0, covariance Σ_0, control inputs u_t, measurements $\bar{q}_t$

Outputs: updated state estimation $\hat{q}_t$, covariance estimation Σ_t

1. Initialize $\hat{q}_{t-1} \leftarrow q_0$
2. Initialize $\Sigma_{t-1} \leftarrow \Sigma_0$
3. For each time step t do:
   (a) ***Prediction step:***
   - Predict mean state: $\hat{q}_t \leftarrow A\hat{q}_{t-1} + B_t u_t$
   - Predict state covariance: $\Sigma_t \leftarrow A\Sigma_{t-1}A^T + R$

   (b) ***Update step:***
   - Compute Kalman gain: $G_t \leftarrow \Sigma_t C_t^T (C_t \Sigma_t C_t^T + Q)^{-1}$
   - Update mean state with measurement: $\hat{q}_t \leftarrow \hat{q}_t + G_t(\bar{q}_t - C_t\hat{q}_t)$
   - Update state covariance: $\Sigma_t \leftarrow (I - G_t C_t)\Sigma_t$

   (c) Prepare for the next iteration:
   - Set $\hat{q}_{t-1} \leftarrow \hat{q}_t$
   - Set $\Sigma_{t-1} \leftarrow \Sigma_t$

estimated as the midpoint $(x^l + x^r)/2$, while the vertical coordinate is given by the average of the two fitted lines at the corresponding endpoints. The final expression is

$$(x, y) = \begin{cases} \left(x^p, y^p + \frac{r_l}{2}\right), & \text{if } \theta = \frac{\pi}{2}, \\ \left(\frac{x^l+x^r}{2}, \frac{k^l x^l + b^l + k^r x^r + b^r}{2}\right), & \text{otherwise.} \end{cases} \tag{11}$$

Now, the RGB-D image-based pose estimation algorithm is summarized in Algorithm 1.

3.3. Pose estimation by fusing odometry data

If object's odometry data can be achieved by the observation platform, a Kalman filter-based pose estimation can be further constructed to improve the robustness and accuracy. The detailed algorithm is shown in Algorithm 2.

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

Table I. *Model performance comparison.*

Model	FLOPs(B)	FPS	Params(M)	mAP_{val}^{50-95}
YOLO9n	7.7	8	2.0	60.4
YOLO10n	6.7	10	2.3	48.4
YOLO11n	6.5	11	2.6	52.3
YOLO12n	6.5	12	2.6	61.6
Replacement 1	1.6	23	0.6	42.0
Replacement 2	4.4	16	0.3	38.5
Replacement 3	4.4	19	0.3	37.5
Ours	1.7	22	0.6	43.1

In this algorithm, definitions of A, B_t, and u_t are given (1). $C_t = I$ is the observation matrix. Σ_0 represents the corresponding covariance matrix of $\hat{q}_0$, which is set as a zero matrix. $\bar{q}_t = [x_t, y_t, \theta_t]^T$ represents the estimation values obtained by Algorithm 1. The covariance matrices R and Q are set as follows:

$$R = \begin{bmatrix} r^x & 0 & 0 \\ 0 & r^y & 0 \\ 0 & 0 & r^\theta \end{bmatrix}, \quad Q = \begin{bmatrix} q^x & 0 & 0 \\ 0 & q^y & 0 \\ 0 & 0 & q^\theta \end{bmatrix}, \tag{12}$$

where the parameters r^x, r^y, r^θ and q^x, q^y, q^θ are estimated by the sample variance of the odometry data and the pose obtained by Algorithm 1, respectively. Here, a pre-experiment sampling process can be conducted to complete the estimation.

4. Simulation and experiment

4.1. Model performance comparison

To highlight the lightness and rapidity of the propose model in this paper, comparative experiments are conducted with the lightweight models in recently proposed YOLO series detection model, specifically the YOLO9n [27], YOLO10n [40], YOLO11n [41], and YOLO12n [42] model. The entire experiments are carried out on a Jetson Nano with 4 GB of memory. Various indicators were used to measure the time complexity, space complexity, and detection accuracy of each model. The specific details are shown in Table I.

From Table I, it is evident that while our model exhibits a marginal accuracy trade-off compared to state-of-the-art algorithms, its parameter count and computational requirements are reduced to only one-fourth of other models. This optimization enables the model to achieve double the inference speed in embedded systems, effectively satisfying real-time application requirements.

To further validate the effectiveness of the proposed architecture, we conduct an ablation study by performing the following module replacements:

1. replacing CSP modules with convolutional layers of equal depth.
2. using a fixed number of channels throughout the backbone instead of gradual channel increment.
3. considering both (1) and (2) simultaneously.

As shown in Table I, the proposed model achieves the highest $mAP_{val}^{50\text{-}95}$ of 43.1 with relatively low computational cost and a compact parameter size, indicating a favorable balance between accuracy and efficiency. In comparison, while replacement (1) has a similar model size, it sacrifices some accuracy. Replacements (2) and (3), in contrast, are more computationally expensive and yet exhibit lower

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

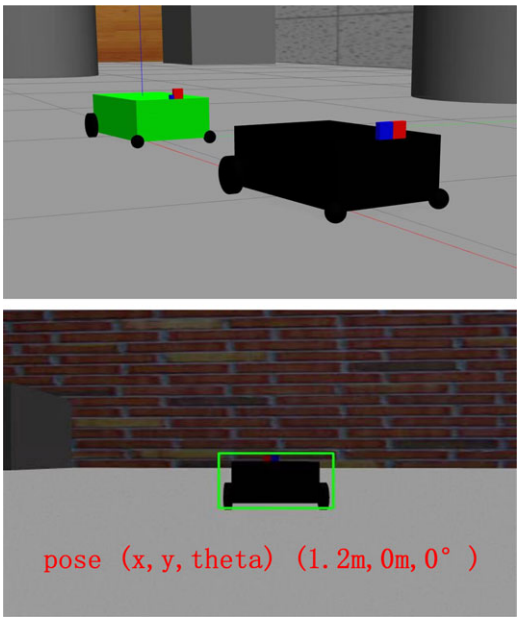


Figure 4. *Simulation in Gazebo with a black leader robot and a green follower robot.*

performance due to the removal of the gradual channel increment. These results demonstrate that the overall architectural design of the proposed model is better suited for lightweight object detection.

4.2. Simulation

First, a numerical simulation that contains two mobile robots with one leader and one follower is conducted in Gazebo shown in Figure 4, where the follower will make the pose estimation for the leader. Initial poses of the leader and follower are, respectively, set as $(1.2m, 0m, 0°)$ and $(0m, 0m, 0°)$. The depth measurement and odometry information are supposed to be disturbed by Gaussian noises that both follow $N(0, 0.1)$. The sampling interval of the RGB-D camera is set to 30 ms. For visual detection, the maximum and minimum detection ranges are set to 5 and 1 m, respectively. The horizontal field of view of the camera is set as 60°.

Let the leader robot, respectively, move along an S-shaped curve and circular curve. Then, under the pose estimation Algorithms 1 and 2, we can obtain the simulation result shown by Figure 5. From Figure 5, we can see that if the pose information can be obtained in real time, both Algorithms 1 and 2 can achieve a good estimation performance, which validates the effectiveness of the algorithms. However, Algorithm 2 will be more robust if the measure error is very big. To further verify this point, mean absolute error (MAE), maximum error, and minimum error are used to evaluate the performance of Algorithms 1 and 2. By comparing both algorithms with the actual values, the error analysis is given in Tables II and III. It can be seen that Algorithm 2 achieves lower MAE (e.g., 0.0068 vs. 0.0258 for x/m in S-shape, 0.0287 vs. 0.1153 for x/m in circular) and smaller maximum errors (e.g., 0.1266 vs. 0.4939 for y/m in S-shape, 0.1124 vs. 0.2924 for y/m in circular) compared to Algorithm 1. Through this analysis, we can see that the Kalman filter based pose estimation by fusing odometry data can achieve better accuracy and robustness, especially under large measurement errors.

4.3. Experimental result

In this part, we will conduct an experiment in actual physical platform to further investigate the performance of our algorithm. The platform contains two intelligent mobile robots depicted in Figure 6. The specific equipment involved in the experiment is shown in Table IV.

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

Table II. *Error analysis of S-shape route.*

	Algorithm 1			Algorithm 2		
Axis	**MAE**	**Max error**	**Min error**	**MAE**	**Max error**	**Min error**
x/m	0.0258	0.2025	0.0004	0.0068	0.0328	0.0001
y/m	0.0854	0.4939	0.0020	0.0174	0.1266	0.0001
$\theta/^\circ$	12.1051	25.3680	0.2040	3.4420	6.3751	0.0297

Table III. *Error analysis of circular route.*

	Algorithm 1			Algorithm 2		
Axis	**MAE**	**Max error**	**Min error**	**MAE**	**Max error**	**Min error**
x/m	0.1153	0.4421	0.0008	0.0287	0.0736	0.0002
y/m	0.1324	0.2924	0.0015	0.0475	0.1124	0.0001
$\theta/^\circ$	16.4087	41.7747	0.2268	5.7067	16.6685	0.0277

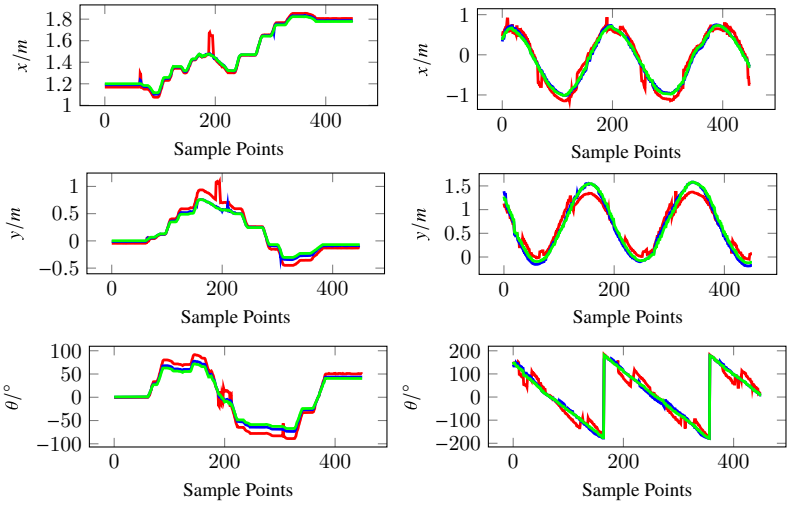


Figure 5. *Pose estimation results for the S-shaped route (left column) and circular route (right column). The red, blue, and green curves represent the results of Algorithms 1, 2, and the actual values, respectively.*

Let the leader robot, respectively, be stationary and move along an S-shaped curve. Then, under Algorithms 1 and 2, we can obtain the experiment result shown by Figure 7. From Figure 7, we can see that the odometry information exists a big error, especially if the robot is moving, while both Algorithms 1 and 2 can achieve a good estimation performance at speeds exceeding 20 FPS. This experiment further validates the effectiveness of the proposed algorithm.

Remark 2. *Due to the uncertainties in the estimation process, such as the measurement error, image quality, estimation errors are inevitable. For example, two vectors* $(x^p - x^l, y^p - y^l)$ *and* $(x^p - x^r, y^p - y^r)$ *are orthogonal theoretically for the left/right yaw case shown in Figure 1, but in practice they may not. Here, the estimation of parameters is computed directly based on the Algorithms 1, 2 with the formular given in Subsection 3.2, and we only adopt the following strategy to improve the robustness and precision during the estimation progress: when the object detection module occasionally fails, the pose estimation is skipped for the current frame and the previous pose is used as a temporary approximation. Other scenarios, such as odometry data loss, have not been considered in our algorithm. Consequently,*

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

Table IV. *Experimental facilities.*

Name	Value
Hardware configuration	Two mobile robots with identical hardware
Controller framework	Jetson Nano and STM32
Operating system	Ubuntu 18.04 + JetPack 4.3
ROS version	Melodic
Encoder resolution	2688 pulses per revolution
Communication method	WiFi module
Depth camera resolution	640 × 480
Frame rate	30 FPS



Figure 6. *Experimental setup.*

inadequate measurement data or low-precision sensor inputs may cause complete malfunction of the proposed algorithmic system.

It is evident that the robustness and accuracy of the algorithm still need to be further improved. Regarding potential countermeasures, one can enhance the image quality by incorporating advanced image enhancement techniques [35, 36, 43–45] and image quality assessment methods [46–48]. Besides, several fault-tolerance strategies and data filtering methods need to be further developed to inhibit the drift or error of the measurement data.

5. Conclusion

In this study, two algorithms have been proposed to solve the real-time pose estimation problem for a class of mobile robots with rectangular body. It should be noted that the algorithm can be easily extended to other kinds of robots with a certain geometry body as long as the pose can be uniquely determined. Both simulation and experiment show that the proposed algorithm can achieve the good performance in terms of real-time performance and accuracy.

Note that Algorithm 2 proposed in this paper depends on the velocity odometry data, which is supposed that it can be directly obtained by the follower. However, in practice, the assumption is hard to be satisfied. In addtion, the environment considered in this study is relatively simple, We have not taken into consideration scenarios that involve obstacles, potential sensor malfunctions, and the filtering of sensor measurement data. The real-time pose estimation algorithm with higher accuracy and robustness that completely depends on the measurement information in complex environment is a direction of our further work.

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

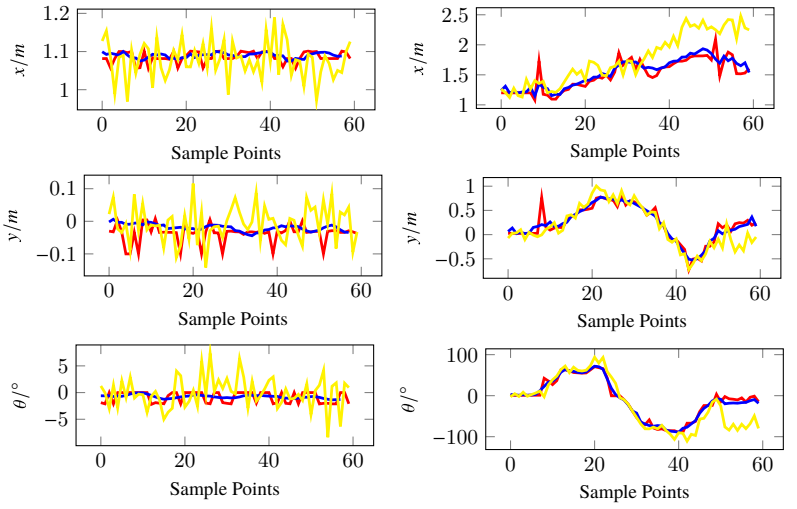


Figure 7. *Pose estimation results for the initial pose with stationary object (left column) and S-shaped route with moving object (right column). The red, blue, and yellow curves represent the results of Algorithms 1, 2, and the odometry, respectively.*

Author contribution. Haoran Tang and Bo Wang conceived the study and designed the methodology. Haoran Tang performed the simulation and experiment. All authors contributed to the review and editing of the manuscript.

Financial support. This work was supported by the National Natural Science Foundation of China (under grants 61903110 and 62073108) and Natural Science Foundation of Zhejiang Provincial (under grant LZ23F030004).

Competing interests. The authors declare no conflicts of interest exist.

References

[1] C. Zhai, P. Zhang, H. Xu, X. Yuan, L. Zhou and R. Wu, "The Application and Inspiration of Robots in the US Military," *2023 8th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)* (IEEE, 2023) pp. 24–28.

[2] H.-Y. Lin, Z.-Y. Xu, J.-Y. Zhou and J.-Y. Chen, "A ROS-based Agricultural AI-Driven AGV (A3GV) with Collaboration and Guiding from Drones in the Outdoor Farming Fields," *2024 IEEE 21st Consumer Communications & Networking Conference (CCNC)* (IEEE, 2024) pp. 1–2.

[3] M. Hassan, J. Eberhardt, S. Malorodov and M. Jäger, "Robust multiview 3d pose estimation using time of flight cameras," *IEEE Sens. J.* **22**(3), 2672–2684 (2021).

[4] X. Chen, Z. Xing, L. Feng, T. Zhang, W. Wu and R. Hu, "An etcen-based motion coordination strategy avoiding active and passive deadlocks for multi-AGV system," *IEEE Trans. Autom. Sci. Eng.* **20**(2), 1364–1377 (2023).

[5] J. Meng, S. Wang, L. Jiang, Z. Hu and Y. Xie, "Accurate and efficient self-localization of AGV relying on trusted area information in dynamic industrial scene," *IEEE Trans. Veh. Technol.* **72**(6), 7148–7159 (2023).

[6] Y. Zhang, B. Li, S. Sun, Y. Liu, W. Liang, X. Xia and Z. Pang, "GCMVF-AGV: Globally consistent multi-view visual-inertial fusion for AGV navigation in digital workshops," *IEEE Trans. Instrum. Meas.* **72**, 1–16 (2023).

[7] G. S. Ramos, M. F. Pinto and D. B. Haddad, "Advancing UAV swarm autonomy with ARCog-NET for task allocation, path planning, and formation control," *Robotica*, 1–45 (2025). doi: 10.1017/S0263574725101914.

[8] W. Ding, Z. Pei, T. Yang and T. Chen, "Dynamic simultaneous localization and mapping based on object tracking in occluded environment," *Robotica* **42**(7), 2209–2225 (2024).

[9] J. Zeng, H. Zhong, Y. Wang, S. Fan and H. Zhang, "Autonomous control design of an unmanned aerial manipulator for contact inspection," *Robotica* **41**(4), 1145–1158 (2023).

[10] C. Liu, K. Shi, K. Zhou, H. Wang, J. Zhang and H. Dong, "Rgbgrasp: Image-based object grasping by capturing multiple views during robot arm movement with neural radiance fields," *IEEE Rob. Autom. Lett.* **9**(6), 6012–6019 (2024).

[11] Y. Zhou, J. Luo and M. Wang, "Dynamic manipulability analysis of multi-arm space robot," *Robotica* **39**(1), 23–41 (2021).

[12] J. Liu and G. Guo, "Vehicle localization during gps outages with extended kalman filter and deep learning," *IEEE Trans. Instrum. Meas.* **70**, 1–10 (2021).

[13] H. Liang, Y. He, C. Zhao, M. Li, J. Wang, J. Yu and L. Xu, "HybridCap: Inertia-aid Monocular Capture of Challenging Human Motions," *AAAI Conference on Artificial Intelligence* (2023) pp. 1539–1548.

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

[14] S. Zhang, B. Qiang, X. Yang, M. Zhou, R. Chen and L. Chen, "Efficient pose estimation via a lightweight single-branch pose distillation network," *IEEE Sens. J.* **23**(22), 27709–27719 (2023).
[15] C. Wang, D. Zhu, L. Sun, C. Han and J. Guo, "Real-time through-wall multiperson 3d pose estimation based on mimo radar," *IEEE Trans. Instrum. Meas.* **12**(8), 10589–10600 (2024).
[16] T. Qin, P. Li and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Rob.* **34**(4), 1004–1020 (2018).
[17] L. Haomin, Z. Guofeng and B. Hujun, "A survey of monocular simultaneous localization and mapping," *J. Comput. Aid Mol. Des.* **28**(6), 855–868 (2016).
[18] J. Cheng, L. Zhang, Q. Chen, X. Hu and J. Cai, "A review of visual slam methods for autonomous driving vehicles," *Eng. Appl. Artif. Intell.* **114**, 104992 (2022).
[19] S. Qiao, H. Zhang, F. Xie and Z. Jiang, "Deep-learning-based direct attitude estimation for uncooperative known space objects," *IEEE Trans. Aerosp. Electron. Syst.* **60**(3), 2526–2541 (2024).
[20] X. Zhang, Z. Jiang and H. Zhang, "Real-time 6d pose estimation from a single rgb image," *Image Vision Comput.* **89**, 1–11 (2019).
[21] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige and N. Navab, "Model Based Training, Detection and Pose Estimation of Texture-less 3d Objects in Heavily Cluttered Scenes," *Computer Vision – ACCV 2012* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013) pp. 548–562.
[22] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton and C. Rother, "Learning 6d Object Pose Estimation Using 3d Object Coordinates," *Computer Vision – ECCV 2014* (Springer International Publishing, Cham, 2014) pp. 536–551.
[23] T. Höfer, F. Shamsafar, N. Benbarka and A. Zell, "Object Detection and Autoencoder-based 6d Pose Estimation for Highly Cluttered Bin Picking," *2021 IEEE international conference on image processing (ICIP)* (IEEE, 2021) pp. 704–708.
[24] J. Josifovski, M. Kerzel, C. Pregizer, L. Posniak and S. Wermter, "Object Detection and Pose Estimation Based on Convolutional Neural Networks Trained with Synthetic Data," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018) pp. 6269–6276.
[25] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 779–788.
[26] J. Redmon and A. Farhadi, Yolov3: An incremental improvement (2018).
[27] C.-Y. Wang, I.-H. Yeh and H.-Y. M. Liao, Yolov9: Learning what you want to learn using programmable gradient information (2024).
[28] Z. Ge, S. Liu, F. Wang, Z. Li and J. Sun, Yolox: Exceeding yolo series in 2021 (2021).
[29] L. Huang, W. Li, Y. Tan, L. Shen, J. Yu and H. Fu, Yolocs: Object detection based on dense channel compression for feature spatial solidification (2024).
[30] J. Song, M. Zhou, J. Luo, H. Pu, Y. Feng, X. Wei and W. Jia, "Boundary-aware feature fusion with dual-stream attention for remote sensing small object detection," *IEEE Trans. Geosci. Remote Sens.* **63**, 1–13 (2024).
[31] S. Cheng, J. Song, M. Zhou, X. Wei, H. Pu, J. Luo and W. Jia, "Ef-detr: A lightweight transformer-based object detector with an encoder-free neck," *IEEE Trans. Ind. Inf.* **20**(11), 12994–13002 (2024).
[32] M. Xu, Y. Wang, B. Xu, J. Zhang, J. Ren, Z. Huang, S. Poslad and P. Xu, "A critical analysis of image-based camera pose estimation techniques," *Neurocomputing* **570**, 127125 (2024).
[33] H. Huang, B. Song, G. Zhao and Y. Bo, "End-to-end monocular pose estimation for uncooperative spacecraft based on direct regression network," *IEEE Trans. Aerosp. Electron. Syst.* **59**(5), 5378–5389 (2023).
[34] Z. Liu, R. Li, S. Shao, X. Wu and W. Chen, "Self-supervised monocular depth estimation with self-reference distillation and disparity offset refinement," *IEEE Trans. Circuits Syst. Video Technol.* **33**(12), 7565–7577 (2023).
[35] M. Zhou, H. Leng, B. Fang, T. Xiang, X. Wei and W. Jia, "Low-light image enhancement via a frequency-based model with structure and texture decomposition," *ACM Trans. Multimedia Comput. Commun. Appl.* **19a**(6), 1–23 (2023a).
[36] M. Zhou, X. Wu, X. Wei, T. Xiang, B. Fang and S. Kwong, "Low-light enhancement method based on a retinex model for structure preservation," *IEEE Trans. Multimedia* **26b**, 650–662 (2023b).
[37] Y. Shen and X. Zhang, "A dynamic slam system with yolov7 segmentation and geometric constraints for indoor environments," *Robotica*, 1–19 (2025). doi: 10.1017/S0263574725101823.
[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition* (2017) pp. 2117–2125.
[39] R. H. Kenney and J. W. McDaniel, "Cooperative navigation of mobile radar sensors using time-of-arrival measurements and the unscented kalman filter," *IEEE Trans. Radar Syst.* **1**, 435–447 (2023).
[40] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han and G. Ding, Yolov10: Real-time end-to-end object detection (2024).
[41] R. Khanam and M. Hussain, Yolov11: An overview of the key architectural enhancements (2024).
[42] Y. Tian, Q. Ye and D. Doermann, Yolov12: Attention-centric real-time object detectors (2025).
[43] Q. Guo, Z. Zhang, M. Zhou, H. Yue, H. Pu and J. Luo, "Image defogging based on regional gradient constrained prior," *ACM Trans. Multimedia Comput. Commun. Appl.* **20**(3), 1–17 (2023).
[44] L. Fan, X. Wei, M. Zhou, J. Yan, H. Pu, J. Luo and Z. Li, "A semantic-aware detail adaptive network for image enhancement," *IEEE Trans. Circuits Syst. Video Technol.* **35**(2), 1787–1800 (2024).
[45] Q. Guo and M. Zhou, "Progressive domain translation defogging network for real-world fog images," *IEEE Trans. Broadcast.* **68**(4), 876–885 (2022).
[46] M. Zhou, X. Lan, X. Wei, X. Liao, Q. Mao, Y. Li, C. Wu, T. Xiang and B. Fang, "An end-to-end blind image quality assessment method using a recurrent network and self-attention," *IEEE Trans. Broadcast.* **69**(2), 369–377 (2022).

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press

[47] M. Zhou, W. Shen, X. Wei, J. Luo, F. Jia, X. Zhuang and W. Jia, "Blind image quality assessment: Exploring content fidelity perceptibility via quality adversarial learning," *Int. J. Comput. Vision* **133**, 3242–3258 (2025).
[48] W. Shen, M. Zhou, J. Luo, Z. Li and S. Kwong, "Graph-represented distribution similarity index for full-reference image quality assessment," *IEEE Trans. Image Process.* **33**, 3075–3089 (2024).

Cite this article: H. Tang, B. Wang, X. Zhou, Z. Han and Q. Lv (2025). "RGB-D image-based real-time pose estimation algorithm for mobile robots with rectangular body", Robotica **43**, 3080–3093. https://doi.org/10.1017/S0263574725102117

https://doi.org/10.1017/S0263574725102117 Published online by Cambridge University Press