Marker-based Lightweight Monocular Visual Localization for Low-Cost Mobile Robot

Haijiang Gao[1], Ziyang Lu[1], Yubin Zhao[1], Xiaofan Li[2], Huaming Wu[3]
[1]School of Microelectronics Science and Technology, Sun Yat-Sen University
Zhuhai, 519082, P. R. China.
Email: {gaohj8, luzy3}@mail2.sysu.edu.cn, zhaoyb23@mail.sysu.edu.cn
[2]School of Intelligent System Science and Engineering, Jinan University
Zhuhai, 519070, P. R. China.
Email: lixiaofan@jnu.edu.cn
[3]Center for Applied Mathematics, Tianjin University
Tianjin, 300072, P. R. China.
Email: whming@tju.edu.cn

2025 IEEE Smart World Congress (SWC) | 979-8-3315-7598-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/SWC65939.2025.00091

***Abstract*—Vision-based localization for some specific areas is significant to low cost mobile robots, since no addition devices or even sensors are required to be deployed. Although image processing methods can be applied, the complex environment, high localization accuracy and lightweight of hardware and algorithm requirements are still challenging. In this paper, we propose a marker-based monocular visual localization method to estimate the distance and yawing angle using a single equipped camera. A simple artificial marker is labeled in the typical indoor place to provide the matching reference points. We develop the adaptive threshold to convert the image into binary, and extract the contours to recognize the position. The distance and angle of the marker is calculated by solving the Perspective-n-Points (PnP) problem. The proposed method is evaluated with different lighting, different floors and positions, where the estimated deviation of distance is 1.8 cm and the estimated deviation of yawing angle is 0.029 radians.**

***Index Terms*—Image processing, low-cost mobile robot, localization, perspective-n-points problem.**

I. Introduction

Mobile robots are widely applied in industry, agriculture, medicine and domestic [1], etc. The ability of detection and localization is significant for mobile robots [2]. For instance, in the automated assembly lines, robots need to estimate the contact points of the objects and perform the tasks of grasping [3]. And in a self-navigation application, robots need to locate their positions to determine the following actions. For domestic robots, the cleaning or carrying services can be achieved with high accurate localization techniques. Existing approaches of robotic localization include wireless local area network (WLAN) based [4], laser rangefinder based [5] and vision-based [6]. For the low cost mobile robot, less sensors are equipped to reduce the overall cost. In this case ,vision-based approaches have the advantages of low-cost and rich visual information, which is compatible for mobile robot.

There are recent researches that focus on applying visual localization for robots [7]. For instance, the robotic grasping system uses the eye-in-hand camera to capture information [8], the modular robots use the RGB-D camera to acquire RGB and depth images of scene [9], and the surgical navigation based on optical tracking system [10]. Feature extraction from the image is essential for pose estimation in vision-based application [11]. A probabilistic paradigm and a set of corresponding 3D-to-2D points [12] are applied to locate the position of robot. These works are mainly used for some specific industry or medical applications.

However, the complex environment and varying lighting will generate noise and degrade the estimated accuracy in visual localization. Firstly, images captured in low lighting usually contain low visibility and low contrast, as well as the image noise and blur. Therefore, it is important to enhance the images for better detection. Ho et al. propose a binarization method based on deep learning technology [13]. Yang et al. use an iterative multi-branch network to extract the feature, and reconstruct the image to enhance the quality of images [14]. On the contrary, there are wrong detection or partial covering of the objects in complex environment. In these scenes, the estimation accuracy is degraded. He et al. propose a generative feature-to-image framework [8]. This framework accepts the features as input and produces the images as output. Iterative process is conducted to eliminate the deviation between approximated image and the real image, to optimize the pose estimated for

Corresponding Author: Yubin Zhao. This work was partially supported by National Nature Science Foundation of China (No. 62271232, 62171484).

2993-396X/25/$31.00 ©2025 IEEE
DOI 10.1109/SWC65939.2025.00091

textureless objects. Liu et al. propose an unsupervised monocular depth estimation framework that combined visual and inertial attributes, to improve the pose estimated accuracy [15]. However, the cumulative error of inertial sensor will leave to a degradation of the accuracy during long-term working. Ding et al. build a database with extracted features of images and the ground truth absolute poses [16]. Walch et al. use the convolution neural network (CNN) to regress the camera pose from a captured image [17]. Yang et al. combine Transformer and CNN modules to improve the performance [18]. However, such models contain a large amount of parameters, and it is difficult to deploy such models in the low-cost and real-time application. In addition, the deep learning based algorithms require large data amount for training, whereas it still can not explore all the possibilities in the complex domestic environment.

Accounting for the above challenges, we propose a lightweight visual localization method based on monocular visual camera and a 2D marker for low-cost mobile robot system. The robot needs no extra electronic devices but just a simple marker label which can be stick to the corner to achieve high accurate real-time localization. Our proposed method contains three parts, which are image processing, marker feature extraction and PnP based localization. The proposed method is robust to the varying lighting and complex environment e.g., different textures of floor plans and walls. And the contributions of our work are summarized as follows:

- In the visual signal processing, we apply an adaptive-threshold method to convert the image into binary. The threshold is calculated according to the neighbor pixels. When the image suffers from different illumination in different regions, the adaptive threshold can reduce the interference and increase the accuracy of detection.
- We propose a grouping method to extract 2D feature points from the marker, which can improve the accuracy in complex environment. The marker contains several triangles and rectangles. Here, the contour simplification based on Douglas-Peucker algorithm and grouping based marker recognition are applied to decline the interference of environment and improve the accuracy of proper matching.
- The localization problem is formulated as the Perspective-n-Points (PnP) problem according to the extracted feature points. We provide the close-form solution by matching the 3D points according to the translation matrix. Therefore, the overall localization complexity is rather low, which can be implemented in the lightweight MCU of the robot.

We evaluate the proposed method on a cleaning robot that equipped with monocular camera. The experiments

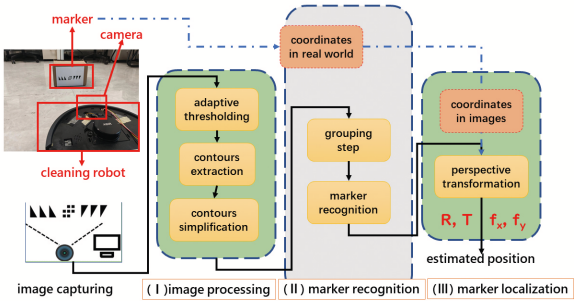


Fig. 1. The architecture of proposed method

with varied illumination, different resolution, different floors and positions show the robustness of our proposed method. In addition, we compare the estimated accuracy with other methods that based on extended Kalman Filter (EKF), the adaptive extended Kalman Filter (AEKF) and RPnP [10], [19], [20]. Experimental results have validated the performance of our proposed method. The mean deviations of estimated distance and angle are 1.8 cm and 0.029 radians, which outperforms other algorithms.

II. System Design

The proposed monocular visual localization scheme contains three parts: image processing, marker recognition, and marker localization, as illustrated in Fig. 1. The whole scheme is implemented in a low cost mobile robot. We print a typical designed marker at a certain place of the room. Such marker contains 6 triangles in two groups and put 7 small rectangles between these two groups. In the image processing, we enhance the contrast of the image and obtain the contour information. Then, we recognize the position of the marker by grouping and filtering the contours. Localization is executed by obtaining the pixel coordinates of feature points, and estimate their positions relative to the camera by solving PnP problem.

A. Image Processing

1) Adaptive Threshold: In the stage of processing, the input image is converted to grayscale using the formula [21]:

$$F = 0.299 \cdot Red + 0.587 \cdot Green + 0.114 \cdot Blue \quad (1)$$

where F is the grayscale image. Then, we convert the image from grayscale to binary. Let $F_{(u,v)}$ be the gray value at the coordinate (u, v) of the image, and $\delta_{(u,v)}$ is the corresponding threshold, the binary image $f_{b(u,v)}$ is obtained by:

$$f_{b(u,v)} = \begin{cases} 0 & F_{(u,v)} > \delta_{(u,v)} \\ 255 & otherwise \end{cases} \quad (2)$$

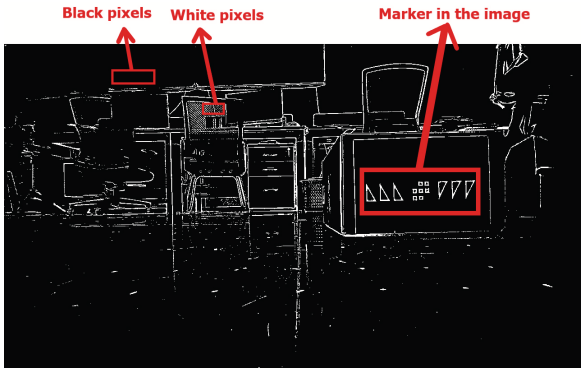


Fig. 2. The binary image. Edges or contours are marked as white color, and the background is black.

In this process, a block of size $k_{size} \times k_{size}$ is used to determine the threshold. Then, we compute the weighted sum of the pixels around (u, v):

$$\delta_{(u,v)} = \sum_{i=u-\frac{k_{size}-1}{2}}^{u+\frac{k_{size}-1}{2}} \sum_{j=v-\frac{k_{size}-1}{2}}^{v+\frac{k_{size}-1}{2}} f_{(i,j)} \cdot G_{(i,j)} - k_{size} \tag{3}$$

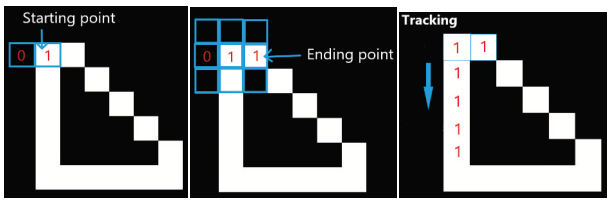
where, $G_{(i,j)}$ is the 2-D Gaussian kernel, with the size of $k_{size} \times k_{size}$. The standard deviation in u and v are σ_u and σ_v, respectively, as follows:

$$G_{(i,j)} = \alpha exp(\frac{-(i-u)^2}{2\sigma_u^2} + \frac{-(j-v)^2}{2\sigma_v^2}) \tag{4}$$

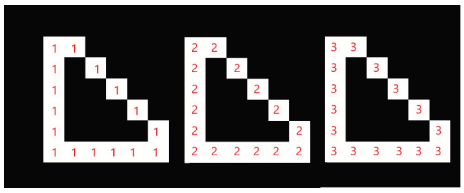
where α is a scale factor with $\sum G_{(i,j)} = 1$.

2) Contours Extraction: The next step is to extract the contours of the marker in order to locate its position in the image. As illustrated in Fig. 2, the black pixels in the binary image are recorded as 0-pixels, and the white pixels are recorded as 1-pixels, respectively. The pixels following method is used to extract contours, where, the 1-pixel is marked as a starting point, if it has one or more 0-pixels in its neighbor region, as shown in Fig. 3(a). In addition, the 1-pixel is marked as an ending point, if it is located in the 3×3 neighborhood of the starting point, as shown in Fig. 3(b). The image is scanned to find the points that satisfy the above conditions. Once a starting point and an ending point are found, continuous procedures are conducted to track the 1-pixels that are connected to the starting point, as illustrated in Fig. 3(c). A contour is found if all connected 1-pixels from the starting point to the ending point are tracked. Their coordinates in the image are recorded, and an index is assigned to distinguish different contours, as presented in Fig. 3(d).

3) Contour Simplification: After extracting all contours in the image, we simplify the contours by removing unnecessary points, but do not change their original



(a) Starting point of the contour (b) Ending point of the contour (c) Tracking and marking the contour



(d) Different contours are detected

Fig. 3. Boundary tracking in the binary image.

shape. For instance, to represent a triangle in the image, we can just record the coordinates of its three vertices and discard others. Eliminating redundant points can not only reduce the processing data, but also simplify the marker detection. Because there are six triangles that contained in the marker, if a simplified contour has three vertices, it can be a part of the marker.

The Douglas-Peucker algorithm is applied to simplify a contour [22]. At the beginning, the first point and the last point of contour are marked as P_1 and P_2, respectively. Here, $\overrightarrow{P_1P_2}$ denotes a line that connects P_1 to P_2. For each point P_i of the contour, the distance between P_i and $\overrightarrow{P_1P_2}$ is calculated. In addition, P_{index} denotes the point that has the maximum distance to the $\overrightarrow{P_1P_2}$, recorded as d_{max}. And P_{index} will be retained if d_{max} is larger than a designated value ϵ, otherwise it will be discarded, as described in Algorithm 1, where $PointList$ can be defined as all or a subset of the points on the contour. By repeatedly applying this algorithm to different subsets of the contour, redundant points are eliminated and the contour is simplified.

B. Marker Recognition

We focus on the triangular contours in the image, and utilize the geometric attributes of the marker to filter out the outliers. As shown in Fig. 4(a), the marker contains three groups of patterns, where both the first group and the third group have three triangles, and the second group have 7 rectangles. Therefore, we take two steps to recognize the position of marker. In the first step, we divide all of the triangular contours into different groups, where, each group has three triangles. In the second step, we analysis the geometric features of each group, to recognize the real group that is belonging to

Algorithm 1 Simplify contours based on Douglas-Peucker Algorithm

Input: The points of contour $PointList$, the threshold ϵ

Output: The points after simplification $OutputList$

Initialize the first point and the last point of the $PointList$: P_1, P_2
Initialize the maximum distance and its index: $d_{max} = 0$, $index = 0$
for all P_i in $PointList$ **do**
  Calculate the distance between P_1 and P_i: $|\overrightarrow{P_1P_i}|$
  Calculate the distance between P_i and $\overrightarrow{P_1P_2}$: $d = |\overrightarrow{P_1P_i}|sin\langle\overrightarrow{P_1P_i}, \overrightarrow{P_1P_2}\rangle$
  Find the maximum distance d_{max} and the corresponding point P_{index}
end for
Compare d_{max} with ϵ
if $d_{max} >= \epsilon$ **then**
  Add P_{index} to $OutputList$: $OutputList \leftarrow P_{index}, P_1, P_2$
end if
return $OutputList$

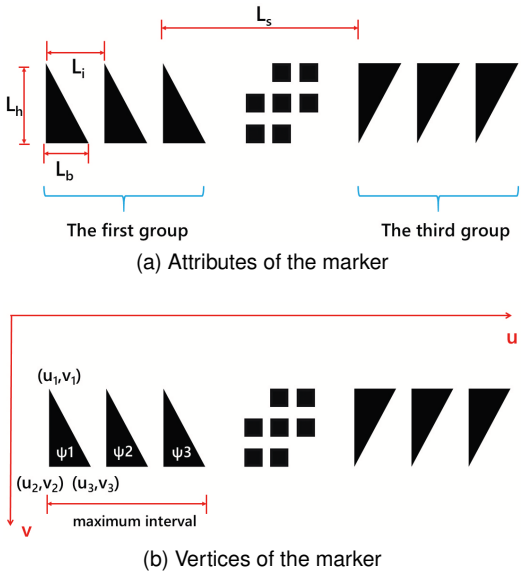


Fig. 4. The geometric attributes of the marker.

the marker.

1) Grouping Step: As illustrated in Fig. 4(a), L_b and L_h are base and height of the triangle, L_i is the interval between two adjacent triangles, and L_s is the space between two groups of triangles. It should be noted that we use uppercase letters, L_b, L_h, L_i and L_s to represent the length in the real world. On the contrary, lowercase letters, l_b, l_h, l_i and l_s are used to denote the length projected in the image, respectively. Let ψ^i $(i = 1, 2, ..., N)$ be the triangles in the image, and $\psi^i.u_j$ $(j = 1, 2, 3)$ be the u-coordinates of vertices, as shown in Fig. 4(b). Then we have:

$$\begin{cases} l_b = |\psi^i.u_1 - \psi^i.u_3| \\ l_i = l_b \cdot \frac{L_i}{L_b} \end{cases} \tag{5}$$

Let l_m be the maximum interval between the triangles within same group. According to 4(b), we have:

$$l_m = 2 \cdot l_i + 3 \cdot l_b \tag{6}$$

where, l_b and l_i are computed by (5). The grouping method based on maximum interval l_m is illustrated in Algorithm 2.

Algorithm 2 Triangles grouping method

Input: The set of triangles $\{\psi^i\}$, the interval and the base of triangles in real world L_i, L_b

Output: The triangles group $\{\boldsymbol{\Delta_i}\}$

for all ψ^{i0} in $\{\psi^i\}$ **do**
  Calculate the interval and the base in image: $l_b = |\psi^{i0}.u_1 - \psi^{i0}.u_3|$, $l_i = l_b \cdot \frac{L_i}{L_b}$
  Calculate the maximum interval: $l_m = 2 \cdot l_i + 3 \cdot l_b$
  Find ψ^{i1} and ψ^{i2} in $\{\psi^i\}$ that are closest to ψ^{i0}
  if $distance(\psi^{i1}, \psi^{i0}) \leq l_m$ and $distance(\psi^{i2}, \psi^{i0}) \leq l_m$ **then**
    Assign ψ^{i0}, ψ^{i1} and ψ^{i2} to the same triangles group $\boldsymbol{\Delta_i}$
  end if
end for
return The triangles group $\{\boldsymbol{\Delta_i}\}$

2) Marker Recognition: Considering that the marker is a predefined pattern, its geometric features are utilized to distinguish with other contours. Therefore, for the three triangles within the same group which is denoted by ψ^1, ψ^2, ψ^3, the deviations in both u-coordinate and v-coordinate are calculated as:

$$\begin{cases} d_u^{ij} = \sum_{n=1}^{3} |\psi^i.u_n - \psi^j.u_n| \\ d_v^{ij} = \sum_{n=1}^{3} |\psi^i.v_n - \psi^j.v_n| \end{cases} \tag{7}$$

A threshold ϵ_d is used to constrain the deviation of d_u and d_v, as described in Algorithm 3, and only the groups with deviations which are smaller than ϵ_d will be recognized as marker.

C. Marker Localization

To further locate the robot based the processed marker image, the perspective projection model is used to compute the relative pose transformation between the camera and marker, as presented in Fig. 6. Let (X_w, Y_w, Z_w) represent the coordinates of marker in the real world, and (X_c, Y_c, Z_c) represent the corresponding coordinates

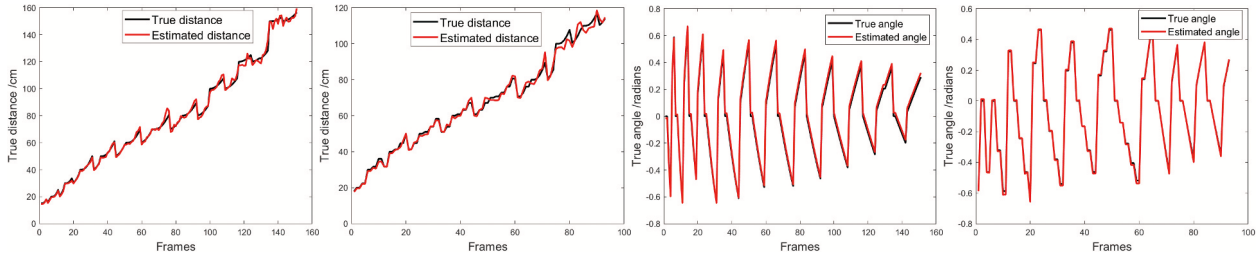


Fig. 5. Experiments of distance estimations under varying lighting condition. (a) Distance estimations with light turn on (b) Distance estimations with light turn off. The ground truth of the distance is gradually increasing over time, to validate the estimated accuracy in different distance between camera and marker.(c) Angle estimations with light turn on (d) Angle estimations with light turn off. The robot is moving left and right repeatedly, to validate the estimated accuracy in different yawing angle

Algorithm 3 Marker recognition

Input: The triangles group $\{\boldsymbol{\Delta_i}\}$

Output: Coordinates of marker in the image $\{(u_i, v_i)\}$

for all $\boldsymbol{\Delta_i}$ in $\{\boldsymbol{\Delta_i}\}$ **do**
  $\psi^1, \psi^2, \psi^3 \leftarrow \boldsymbol{\Delta_i}$
  Calculate the deviations in u-coordinate:
  $d_u^{12} = \sum_{n=1}^{3} |\psi^1.u_n - \psi^2.u_n|$
  $d_u^{23} = \sum_{n=1}^{3} |\psi^2.u_n - \psi^3.u_n|$
  Calculate the deviations in v-coordinate:
  $d_v^{12} = \sum_{n=1}^{3} |\psi^1.v_n - \psi^2.v_n|$
  $d_v^{23} = \sum_{n=1}^{3} |\psi^2.v_n - \psi^3.v_n|$
  Define the constraint of deviations:
  $\epsilon_d = 1.2 \cdot |\psi^2.v_1 - \psi^2.v_2|$
  Compare the deviations with ϵ_d:
  if $d_v^{12} < \epsilon_d$ and $d_v^{23} < \epsilon_d$ and $|d_u^{12} - d_u^{23}| < min(d_u^{12}, d_u^{23})$ **then**
    Record the position of ψ^1, ψ^2,ψ^3 in image: (u_i, v_i)
  end if
end for
return Coordinates of marker in the image $\{(u_i, v_i)\}$

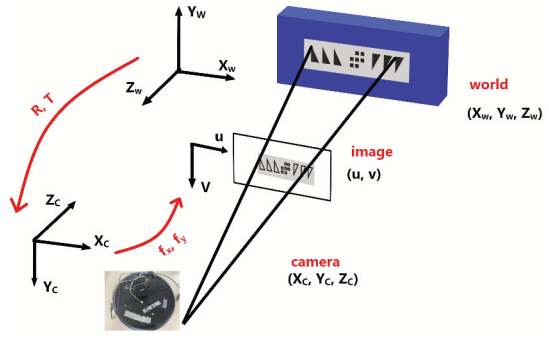


Fig. 6. Perspective projection model.

in the camera frame. The transformation between real world and camera frame can be described by a rotation matrix $\boldsymbol{R}$ and a translation matrix $\boldsymbol{T}$:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \boldsymbol{T} \tag{8a}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{8b}$$

The transformation between the camera frame and the image coordinate can be described by a pin hole camera model [23]. Let (u, v) represent the coordinates of the marker projected in the image, we have:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \tag{9}$$

where f_x and f_y are the focal lengths of the camera in the x and y, u_0 and v_0 account for the translation from the origin of image coordinates to the origin of camera frame. Combining equations (8b) and (9), the homogeneous matrix is written as:

$$\begin{bmatrix} \boldsymbol{A_1} & \boldsymbol{A_2} & \boldsymbol{A_3} \end{bmatrix} \boldsymbol{S} = \boldsymbol{A}\boldsymbol{S} = \boldsymbol{0} \tag{10}$$

where

$$\boldsymbol{S} = [r_{11}, r_{12}, r_{13}, T_x, r_{21}, r_{22}, r_{23}, T_y, r_{31}, r_{32}, r_{33}, T_z]^T \tag{11}$$

and $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, $\boldsymbol{A_3}$ are defined as follows:

$$\boldsymbol{A_1} = \begin{bmatrix} f_x \cdot X_w & f_x \cdot Y_w & f_x \cdot Z_w & f_x \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{12a}$$

$$\boldsymbol{A_2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ f_y \cdot X_w & f_y \cdot Y_w & f_y \cdot Z_w & f_y \end{bmatrix} \tag{12b}$$

$$\boldsymbol{A_3} = \begin{bmatrix} (u_0 - u) \cdot X_w & (u_0 - u) \cdot Y_w & (u_0 - u) \cdot Z_w & (u_0 - u) \\ (v_0 - v) \cdot X_w & (v_0 - v) \cdot Y_w & (v_0 - v) \cdot Z_w & (v_0 - v) \end{bmatrix} \tag{12c}$$

Note that a pair of corresponding points, (u, v) and (X_w, Y_w, Z_w), can provide two equations in (10). Therefore, for each detected group of triangles in the image, there are 9 feature points that can be utilized to calculate the transformation. We solve the PnP problems by matching the corresponding feature points, and find the translation matrix $\boldsymbol{T} = [T_x, T_y, T_z]$ from equation (10). Then, the distance between camera and the marker is equal to $|T_x^2 + T_z^2|$, and the yawing angle is equal to $\arctan(T_x/T_z)$.

III. Experiment

To evaluate the proposed model, we deploy the localization scheme into a vacuum cleaning robot equipped with singular camera, and we also put the marker as the reference point. The localization scheme is compiled in a virtual environment of Linux, and then it is transferred to the robot. During experiments, we put the marker in different positions, and use the camera to capture the images. The ground truth of the physical distance and angle are measured simultaneously. And the accuracy is evaluated by computing the deviation of distance and angle between estimated value and ground truth.

A. Varying Lighting

Firstly, we evaluate the performance with varying lighting condition. The marker is firstly put in the laboratory with the light turn on. During the experiment, we change the relative position of the marker to the robot and estimate its position from the captured images. After testing in 153 frames, we turn off light and repeat the above procedures again. We test the performance in 122 frames with the light turn off. The estimated distance and angle are compared with the ground truth, as shown in Fig. 5. The numerical results are summarized in Table I, where, the mean and maximum deviations are similar under different lighting. However, the number of detected frames decreases when turning off the light. This is caused by a dark environment, which will make it harder to recognize the marker in images. Note that, the estimation error looks similar for both cases. The mean distance error is 1.41 cm when the light is turned on, while it is 1.45 cm when the light is turned off. However, the angle estimation error is even smaller with the average value of 0.0058 radians in the dark which is lower than 0.02 radians for the bright environment. This is due to the less light noise and blur when the detected frame in the dark leads to high accurate estimations. In addition, if the frame is not detectable, the estimation fails.

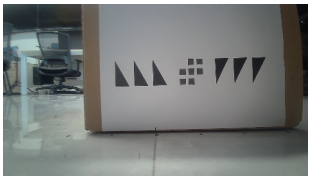
B. Different Floor Textures

In the second experiment, we verify the adaptability of proposed method to different materials of floors. The experimental floors include carpet, wooden and ceramic, as shown in Fig. 7. Figure 8(a) presents the deviation of estimated distance, where, the experiment on ceramic floor has the minimum deviation which is 1.41 cm compared with other two floors. Figure 8(b) presents the deviation of estimated angle, and the estimation on wooden floor has the maximum deviation which is 0.0189 radians. Although the estimation error for different textures is quite close, we find that the white of ceramic floor is easily filtered during the imaging processing stage, which demonstrates the lowest distance estimation error. However, such impact can not affect the angle estimation if the contour is recognized. Thus, the angle estimation performances among different textures are similar.

TABLE I
Mean and maximum deviations (centimeters and radians) in different lighting condition

Lighting	Number of frames		Distance (cm)		Angle (radians)	
	Total frames	Detected frames	Mean	Maximum	Mean	Maximum
turn on light	153	151	1.4106	7.763	0.0214	0.0355
turn off light	122	93	1.4518	6.339	0.0059	0.0237

(a) Experiments on the carpet (b) Experiments on the wooden floor

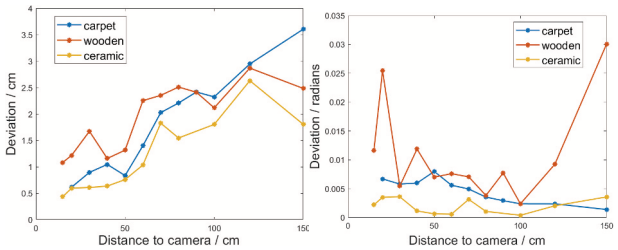


(c) Experiments on the ceramic floor

Fig. 7. Experiments on different floors.

C. Image Resolution Evaluation

The resolution of each frame determines the processed data amount and further the processing time. Here, we evaluate our proposed method with different image resolution. We change the configuration of the camera and capture images with different resolution of 1920×1080, 1280×720 and 640×360 pixels. The experimental result is illustrated in Fig. 10. In the images of low resolution,



(a) Deviation of estimated distance on different floor

(b) Deviation of estimated angle on different floor

Fig. 8. The deviation of estimated distance and angle on different floor. Horizontal axis is the distance between marker and camera.

only part of the marker is recognized, as illustrated in Fig. 9. When the distance between the marker and the camera increases, the estimated deviation also increases. In addition, applying a higher resolution can reduce the deviations of distance, as shown in Fig. 10(a). Moreover, when the distance to camera increases to 60 cm, the deviation of angle increases to 0.024 radians, as shown in Fig. 10(b).

D. Performance Comparison

In addition to our proposed localization method of PnP problem, we also compare with other methods, including EKF [19], AEKF [20], RPnP [10], and DLT+EKF [24], [25]. These methods utilize the system's dynamic model and the error covariance to predict the pose from noisy signals, and have been employed in the estimation of robot's pose. We collect 490 images that contained different pose of the robot, to calibrate the covariance of measurement and noise. During experiments, the robot is approaching to the marker. We establish a serious of checkpoints along the moving path of the robot, and gauge the ground truth of position using a measuring tape. Images are captured when the robot moves to the checkpoints, and we apply EKF, AEKF, DLT+EKF and our proposed method to estimated the pose from the images, respectively. We compare the estimated results with ground truth and compute the deviation. The experimental results are presented in Fig. 11(a). When $frames = 50$ and $frames = 150$, EKF and AEKF fail to track the marker properly, because the movement of the robot suddenly changes. However, the proposed method can properly track the marker with low error. The cumulative frequency of estimated deviation is illustrated in Fig. 11(b), where the proposed method shows better performance than EKF and AEKF. The result of the experiment with distance from 10cm to 180cm are summarized in Table II, where the proposed method has the mean deviation of 1.8 cm and maximum deviation of 19.8155 cm. Although DLT+EKF is close to our method and the mean deviation is 0.2 cm less than ours, DLT+EKF requires prior statistical information of the environment and our method is much simpler without such prior knowledge, which is robust to different environments.

TABLE II
MEAN AND MAXIMUM DEVIATIONS (CENTIMETERS AND RADIANS) COMPARE WITH DIFFERENT METHODS

	Distance (cm)		Angle (radians)	
Methods	Mean	Maximum	Mean	Maximum
EKF	2.9159	48.7748	0.02924	0.19085
AEKF	4.5644	96.4421	0.02913	0.19085
RPnP	3.1243	33.908	0.0284	0.1904
DLT+EKF	1.6207	20.6617	0.02909	0.19078
Proposed Method	1.8001	19.8155	0.02902	0.18839

IV. CONCLUSION

A visual localization method for low-cost mobile robot is proposed based on the 2D marker and monocular camera. The marker is added to the environment for robust localization. We convert the captured image into grayscale, and extract the geometric features. The angles and distances of marker relative to camera is calculated by matching the corresponding feature points. We evaluate our proposed method in different experiments with challenges of varying illumination and complex environment. Experiments demonstrate that the proposed method is accurate to centimeter level, with the mean distance deviation of 1.4 cm and mean angle deviation of 0.02 radians.

REFERENCES

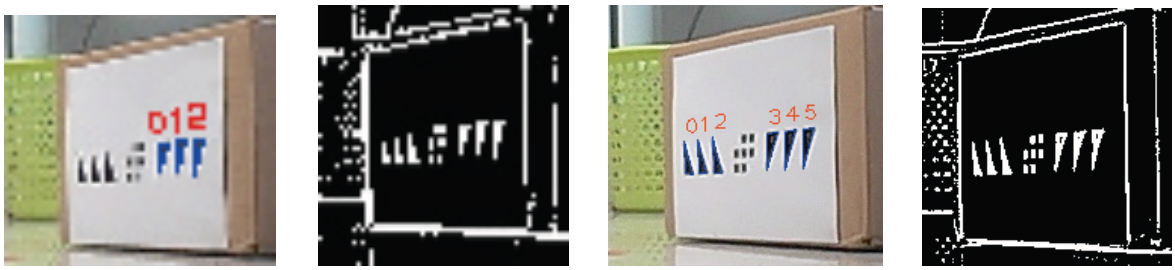
[1] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, Part B, pp. 6019–6039, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821000550

[2] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1677–1734, 2021.

[3] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.

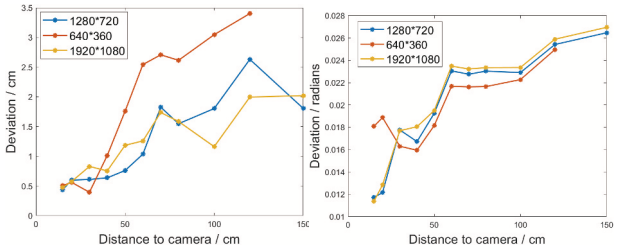
[4] N. A. K. Zghair and A. S. Al-Araji, "A one decade survey of autonomous mobile robot systems," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, p. 4891, 2021.

[5] G. Li, J. Meng, Y. Xie, X. Zhang, L. Jiang, and Y. Huang, "An improved observation model for monte-carlo localization integrated with reliable reflector prediction," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2019, pp. 972–977.



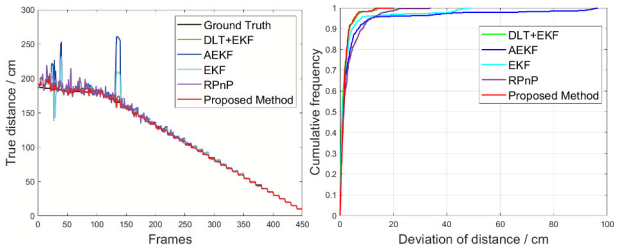
(a) The recognition with resolution of 640×360 (b) Binary image with resolution of 640×360 (c) The recognition with resolution of 1920×1080 (d) Binary image with resolution of 1920×1080

Fig. 9. In low resolution images, only part of the marker is recognized.



(a) Deviation of estimated distance in different resolution (b) Deviation of estimated angle in different resolution

Fig. 10. The deviation of estimated distance and angle in different resolution. Horizontal axis is the distance between marker and the camera.



(a) The estimated position compare with ground truth and other methods (b) The cumulative frequency of different methods

Fig. 11. The experiment results compare with other methods. Our proposed method shows better performance than other methods.

[6] R. Liu, Z. Deng, Z. Cao, M. Shalihan, B. P. L. Lau, K. Chen, K. Bhowmik, C. Yuen, and U.-X. Tan, "Distributed ranging slam for multiple robots with ultra-wideband and odometry measurements," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 684–13 691.

[7] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[8] Z. He, M. Wu, X. Zhao, S. Zhang, and J. Tan, "A generative feature-to-image robotic vision framework for 6d pose measurement of metal parts," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 3198–3209, 2021.

[9] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9626–9633.

[10] J. Wang, T. Zhang, Z. Zhang, M. Q.-H. Meng, and S. Song, "Tracking-by-registration: A robust approach for optical tracking system in surgical navigation," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[11] M. Kalaitzakis, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Experimental comparison of fiducial markers for pose estimation," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 781–789.

[12] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Ep n p: An accurate o (n) solution to the p n p problem," *International journal of computer vision*, vol. 81, pp. 155–166, 2009.

[13] C. C. Ho and C.-D. Lin, "Pose-based visual servoing with lightweight deep-learning binarization for autonomous mobile robot application," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2023, pp. 1093–1099.

[14] S. Yang, D. Zhou, J. Cao, and Y. Guo, "Rethinking low-light enhancement via transformer-gan," *IEEE Signal Processing Letters*, vol. 29, pp. 1082–1086, 2022.

[15] F. Liu, M. Huang, H. Ge, D. Tao, and R. Gao, "Unsupervised monocular depth estimation for monocular visual slam systems," *IEEE Transactions on Instrumentation and Measurement*, 2023.

[16] M. Ding, Z. Wang, J. Sun, J. Shi, and P. Luo, "Camnet: Coarse-to-fine retrieval for camera re-localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2871–2880.

[17] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 627–637.

[18] L. Yang, C. Zhang, G. Liu, Z. Zhong, and Y. Li, "A model for robot grasping: Integrating transformer and cnn with rgb-d fusion," *IEEE Transactions on Consumer Electronics*, 2024.

[19] F. Janabi-Sharifi and M. Marey, "A kalman-filter-based method for pose estimation in visual servoing," *IEEE transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.

[20] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation," in *2017 IEEE power & energy society general meeting*. IEEE, 2017, pp. 1–5.

[21] H. Ayunts and S. Agaian, "No-reference quality metrics for image decolorization," *IEEE Transactions on Consumer Electronics*, 2023.

[22] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[23] P. Sturm, "Pinhole camera model," in *Computer Vision: A Reference Guide*. Springer, 2021, pp. 983–986.

[24] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[25] X. Wu, E. Zakeri, and W.-F. Xie, “Adaptive robust kalman filter for vision-based pose estimation of industrial robots,” in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 2019, pp. 298–302.