Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7299309, 11 pages
https://doi.org/10.1155/2022/7299309

Research Article

A UAV Visual Relocalization Method Using Semantic Object Features Based on Internet of Things

Maolin Wang,[1,2] Hongyu Wang,[3] Zhi Wang,[1,2] and Yumeng Li[4]

[1]*Zhe Jiang Key Laboratory of General Aviation Operation Technology, Jiande, China*
[2]*CAAC Key Laboratory of General Aviation Operation, Department of General Aviation, Civil Aviation Management Institute of China, Beijing, China*
[3]*University of Science and Technology Beijing, Beijing, China*
[4]*Beihang University, Beijing, China*

Correspondence should be addressed to Yumeng Li; liyumeng@buaa.edu.cn

Received 17 November 2021; Accepted 30 December 2021; Published 11 February 2022

Academic Editor: Xiaojie Wang

Unmanned Air Vehicle (UAV) has the advantages of high autonomy and strong dynamic deployment capabilities. At the same time, with the rapid development of the Internet of Things (IoT) technology, the construction of the IoT based on UAVs can break away from the traditional single-line communication mode of UAVs and control terminals, which makes the UAVs more intelligent and flexible when performing tasks. When using UAVs to perform IoT tasks, it is necessary to track the UAVs' position and pose at all times. When the position and pose tracking fails, relocalization is required to restore the current position and pose. Therefore, how to perform UAV relocalization accurately by using visual information has attracted much attention. However, the complex changes in light conditions in the real world have brought huge challenges to the visual relocalization of UAV. Traditional visual relocalization algorithms mostly rely on artificially designed low-level geometric features which are sensitive to light conditions. In this paper, oriented to the UAV-based IoT, a UAV visual relocalization method using semantic object features is proposed. Specifically, the method uses YOLOv3 as the object detection framework to extract the semantic information in the picture and uses the semantic information to construct a topological map as a sparse description of the environment. With prior knowledge of the map, the random walk algorithm is used on the association graphs to match the semantic features and the scenes. Finally, the EPnP algorithm is used to solve the position and pose of the UAV which will be returned to the IoT platform. Simulation results show that the method proposed in this paper can achieve robust real-time UAVs relocalization when the scene lighting conditions change dynamically and provide a guarantee for UAVs to perform IoT tasks.

1. Introduction

Unmanned Air Vehicle (UAV) has the advantages of high autonomy and strong dynamic deployment capabilities [1], which can collect various types of observation data and perform operations tasks accurately. Currently, UAVs mainly conduct one-to-one communication with the control terminal, which satisfies the basic needs of humans for manipulating UAVs to perform some simple tasks [2, 3]. However, in actual applications, data monitoring or operations are often not carried out on a single line, because single-line communication restricts the application performance and benefits of UAVs to a certain extent. The rapid development of the Internet of Things (IoT) has become the mainstream of the times. The IoT consisting of UAVs acting as device nodes can solve the above problems and make the UAVs more intelligent and flexible when performing tasks [4]. The localization problem is a very critical issue in the field of UAVs. The accurate self-localization capability and robustness to common interference factors must be shown when the UAV performs IoT tasks such as industrial inspections and data collection [5, 6]. UAVs are generally localized through the Global Navigation Satellite Systems and Inertial Navigation Systems (INS). When the UAV is working in inclement weather or an indoor environment, the Global Positioning System (GPS) signal may be weak or interrupted

[7]. At this moment, the UAV can rely on the INS system to perform localization within a short period time. However, the localization error of the INS system will accumulate over time; therefore, how to use other supplementary information to help UAVs achieve relocalization and ensure the correct execution of IoT tasks is a very realistic problem. With the development of computer vision, the visual scene matching technology based on camera sensors is becoming increasingly mature, which can provide high-quality services for autonomous UAV flight during IoT tasks [8]. The camera sensor can provide abundant online environmental information and has the advantages of low cost, small size, light weight, noncontact, etc. Additionally, a suitable visual localization system can be combined with GPS and INS systems as a supplementary localization system for UAVs [9, 10].

The UAV needs to track the position and pose at all times during the working process. When the position and pose tracking fails, the relocalization algorithms should be used to restore the current position and pose [11]. The traditional visual relocalization algorithms mostly rely on the similarity among environmental appearances, using artificially designed low-level geometric visual features as the basis for calculating image similarities [12] and then using feature matching to complete scene matching, which can achieve good performance in an environment with constant light conditions. The bag-of-words model was proposed in [13], which uses the K-Means algorithm to construct a "dictionary" and then merge similar features through a clustering approach, utilizing the bag-of-words model library DBoW3 to extract ORB features [14]. The image will be converted into a visual bag-of-words representation vector; then they count the frequency of each component in the vector and calculate the similarity between images through the distance between the representation vectors to complete visual relocalization, where hamming distance or cosine distance may be chosen according to different descriptors. In [14], the authors proposed the FAB-MAP, which is a probabilistic method based on appearance and an extended model of the bag-of-words model. It is based on the Chow-Liu Tree theory to fit the discrete probability distribution, which can well solve the perceptual deviation problem. These two relocalization methods utilize features obtained from the appearance of the images for scene matching. However, in the real world, there are often complex environmental changes such as lighting condition changes, weather changes, and seasonal changes, which lead to a part of key features being strengthened or weakened and a decrease in the accuracy of feature matching. At this time, the relocalization effect of these two methods will be greatly deteriorated, and if we want to keep the performance of them, we have to pay expensive map maintenance costs [15].

In the field of visual localization, SIFT and SURF [16–19] are the two most important local feature descriptors, which are invariant to image rotation and scale scaling, and have a good tolerance for illumination changes and fine-tuning of viewing angles. The use of SIFT and SURF for feature matching can achieve high accuracy, but the computing speed of them is far inferior to the ORB feature. Because of the low-performance processor of the UAV, the above two feature description methods are not compatible with UAV real-time visual localization services [20]. Therefore, how to complete accurate, fast, and robust visual relocalization when UAVs perform IoT tasks is a very challenging problem.

In recent years, deep learning has promoted huge breakthroughs in the field of object detection, and a large number of excellent-performance object-detection schemes have been successively proposed such as Mask R-CNN [21], Faster R-CNN [22], and YOLO [23]. The object-detection method can segment the foreground and background and obtain the semantic information of the objects (such as the categories and attributes) in the scene picture. Compared with geometric visual features, this kind of object-level semantics belongs to high-level features, which is sparse and highly invariant to lighting changes [24, 25]. There is always a semantic gap between using low-level features to describe images and perceiving images by humans. The introduction of semantic features into maps [26–29] makes the description of images closer to the level of human understanding, which can alleviate this problem to a certain extent and improve the robustness of UAV visual relocalization. In [28], the authors indicate that a defined hierarchical structure of semantic information enables improved map reproduction. In [29], the authors introduce semantic information into the map to enable the UAVs to complete terrain classification and navigation.

Oriented to the UAV-based IoT, this paper proposes a UAV visual relocalization method using semantic object features. Firstly, we use YOLOv3 as an object-detection framework to obtain higher-level semantic features that are stable to change in lighting conditions. To describe images, the semantic information is used to abstract the images into the form of topological graphs, which can simplify the preservation and comparison process of the environmental information of images. With prior knowledge of the map, the random walk algorithm [30, 31] is used on the association graphs to match the semantic features and the scenes. Finally, the EPnP algorithm [32] is used to solve the UAV's position and pose for robust UAV relocalization and guarantee of UAV performing IoT tasks.

2. The UAV-Based IoT Model

As the device nodes of the IoT, UAVs can effectively improve the inflexible topological structure of the IoT and are very suitable for data collection and monitoring. Location information is very important during the work of the UAVs. The main components of the UAV-based IoT [4] include the UAV part, the IoT platform, and the communication links (A2G/G2A links). The control commands issued by the IoT platform are uploaded to UAVs via the G2A links, and UAVs perform corresponding tasks according to these commands. When the tracking of position and pose of a UAV fails, the relocalization algorithm is executed. After the UAV completes its relocalization, the position and pose information will be returned to the IoT platform via the A2G link, and the IoT platform will make corresponding adjustments to the tasks of UAVs based on this information [33, 34]. The UAV-based IoT model is shown in Figure 1.

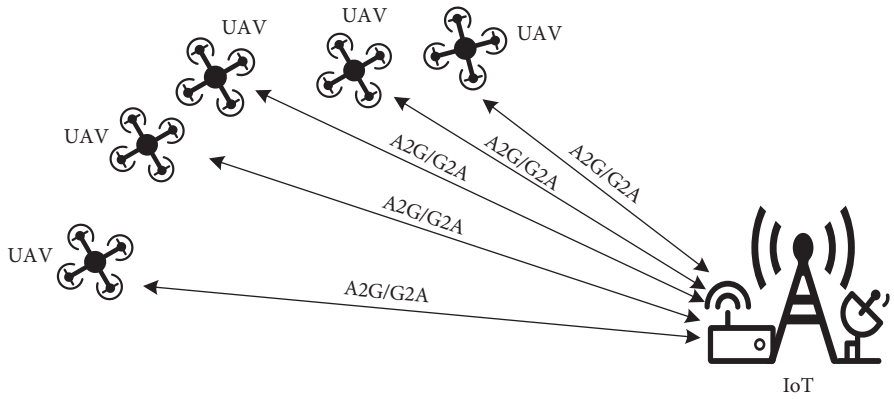


Figure 1: The UAV-based IoT model.

3. Graph Matching Problem Formulation

How to complete scene matching is the key issue in UAV visual relocalization. We abstract the scene images into the form of topological graphs and then transform the scene matching problem into a graph matching problem. The purpose of graph matching is to determine the mapping relationship between the nodes of two graphs. Suppose that the two graphs to be matched are $G^M = (V^M, E^M)$ and $G^N = (V^N, E^N)$, respectively, where V represents the set of nodes in the graph, E represents the set of edges, and c^N and c^M are the numbers of nodes in G^M and G^N, respectively. The weight matrix W of a graph measures the degree of matching between candidate correspondences whose diagonal elements are the matching weights between nodes, and off-diagonal elements represent the matching weights between edges. High weight means that the pair of points or the pair of edges corresponding to the weight have a high matching degree. $X' \in \{0,1\}^{c^M \times c^N}$ is a $c^M \times c^N$-dimensional assignment matrix with one and only one element of 1 in each row and each column. Its elements contain the matching relationship with the nodes of G^M and G^N; namely, $X'_{ia} = 1$ represents that node i in G^M corresponds to node a in G^N; otherwise, $X'_{ia} = 0$. $X = \mathrm{vec}(X')$ is the result of column vectorization of X'; then the graph matching problem can be expressed as an integer quadratic programming problem, which can be written as

$$\begin{aligned} &X' = \arg\max\left(X^T W X\right), \\ &\text{s.t. } X \in \{0,1\}^{c^M \times c^N}, \quad \forall i \sum_{a=1}^{c^M} X_{ia} \le 1, \forall i \sum_{i=1}^{c^N} X_{ia} \le 1. \end{aligned} \tag{1}$$

Graph matching based on random walk [30] is a robust graph matching algorithm for outliers and deformations. The algorithm implements antinoise graph matching by iteratively updating and mining the confidence of candidate matching pairs. The random walk algorithm can give weight to the nodes in the graph, and the association graph can express the matching relationship between two graphs. By transforming the two graphs to be matched into an association graph, the weight of each matching pair in the original two graphs can be obtained, to select the best matching pairs. To select reliable nodes on the association graph, the graph matching problem is modeled as a random walk model, and the process of random walk on the association graph is regarded as a Markov random process, and the weight matrix is used to construct the transfer matrix. Let the weight matrix be W; then the probability transition matrix $P = D^{-1}W$, where D is a diagonal matrix, and the weight matrix W is a symmetric matrix. The weight matrix W can be normalized by multiplying the inverse matrix of D on the left. We denote $X^{(t)T}$ as the probability distribution of all nodes which the random walker may reach on the association graph at time t; then the Markov chain of the random walk process can be expressed as

$$X^{(t+1)T} = X^{(t)T} P. \tag{2}$$

The association graph is a weight undirected graph. By randomly walking on the association graph until the probability distribution of the nodes on the association graph converges, the assignment matrix X' can be obtained, and the matching relationship between G^M and G^N can be determined.

4. Relocalization Model

The traditional UAV visual relocalization methods often achieve scene matching directly calculating the similarity between the input image of the camera and the image in the map library, while the method we proposed in this paper abstracts images as semantic topology graphs and indirectly completes the calculation of the image similarity by comparing the structures of semantic topology graphs. The UAV visual relocalization model (see Figure 2) mainly includes three modules: image processing and representation module, scene matching module, and relocalization module.

The relocalization process is divided into three steps:

(i) Firstly, the current scene is captured by the UAV camera, and the YOLOv3 object detection framework in the image processing module is used to obtain the semantic information of the object

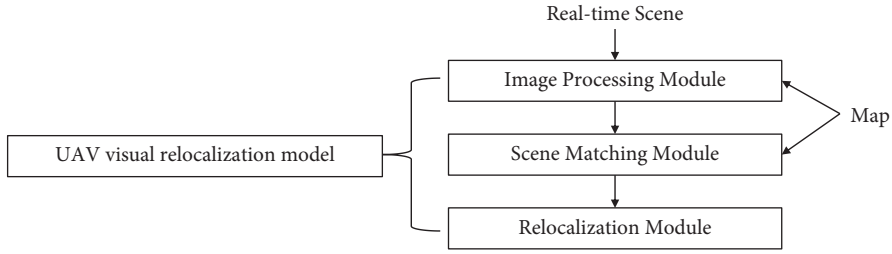


Figure 2: Relocalization model.

contained in the scene image (including the semantic labels and the pixel position information of each object in the image), and then we regard the semantic labels of objects as the nodes of semantic topology graphs. A scene map library is established as an internal representation of the environment for comparison with the real-time image captured by the UAV camera.

(ii) Secondly, the association graphs are generated based on the common semantic information of the images to be matched. We comprehensively considered the semantic information difference, node position difference, and topological structure difference of the images to be matched. The random walk algorithm is used to complete the mapping of semantic feature pairs; afterwards, we can calculate the similarity between the input image and the image in the map library and complete the scene matching of UAV.

(iii) Finally, according to the scene matching result and the information stored in the map library, the 3D position information of the semantic feature points in the world coordinate system can be obtained. Combining the 2D position information of the semantic feature points we have known in the camera coordinate system, we apply the EPnP algorithm to solve the camera position and pose corresponding to the motion of the above 3D points to 2D points to realize the visual relocalization of the UAV.

4.1. Image Processing and Representation. YOLOv3 is a one-stage object detection algorithm. It is the third version of the YOLO series that skips the step of generating candidate regions and directly extracts features from the network to predict the classification and position of objects. The Darknet-53 network structure is adopted in YOLOv3, and there are 53 convolutional layers of size 1×1 or 3×3, as shown in Figure 3.

Compared with the two-stage object detection algorithms of the R-CNN series, YOLOv3 has an obvious speed advantage, which can well serve the real-time detection works, and has high adaptability to the real-time scene matching task of UAVs. The effect of object detection using UAV with YOLOv3 is shown in Figure 4(a). The outputs of object detection are saved in text form in the format of

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3/2$	128×128
$1\times$	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	$3 \times 3/2$	64×64
$2\times$	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	$3 \times 3/2$	32×32
$8\times$	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	$3 \times 3/2$	16×16
$8\times$	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024	$3 \times 3/2$	8×8
$4\times$	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3: The network structure of YOLOv3.

[object semantic labels, upper left pixel coordinate values of the detection frame, lower right pixel coordinate values of the detection frame], as shown in Figure 4(b).

4.2. Scene Matching. The scene matching module is the core of realizing the UAV visual relocalization. Suppose that the two images to be matched are G^M and G^N, respectively, and $G_{\text{ass}}^{M,N}$ is their association graph. v_i^M and v_j^M denote the nodes of G^M, and e_{ij}^M denotes their edge. v_a^N and v_b^N denote the nodes of G^N, and e_{ab}^N denotes their edge. The UAV visual scene matching method based on semantics object features proposed in this paper comprehensively considers the semantic information difference, node position difference, and topological structure difference of the graphs to be matched in the following ways:

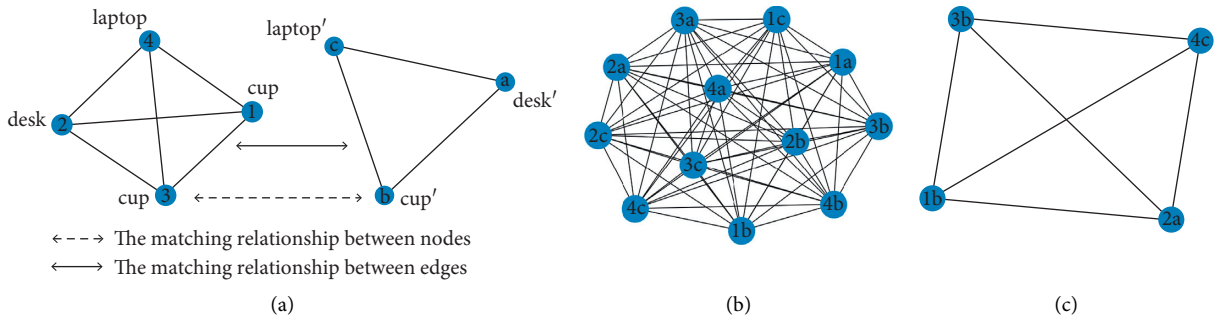
(1) Semantic information difference: a new node $v_{ia}^{M,N}$ in $G_{\text{ass}}^{M,N}$ will be generated by a pair of nodes v_i^M and v_a^N with the same semantic label, while a pair of nodes with different semantic labels will be defined as a conflict matching pair and will not appear in the association graph, which simplifies the structure of the association graph (see Figure 5) and also enables

(a)

cup 1403 467 1758 1311
bottle 2915 1170 3132 1735
bottle 2974 1087 3237 1708
bottle 2374 1127 2629 1721
bottle 2688 1123 2845 1607
book 2442 1866 2526 2539
bottle 2751 1333 2934 1755
book 11 506 256 1336
book 29 310 463 1333
cup 1852 802 2245 1288
book 2516 1881 2634 2519
book 2330 1853 2437 2569
bottle 3195 1065 3556 1707
bottle 2917 1110 3158 1742
laptop 912 1886 1959 3215
mouse 2144 2889 2330 3159
chair 486 4154 2048 5140

(b)

FIGURE 4: The effect of object detection using UAV with YOLOv3. (a) The output in image form. (b) The output in text form.



(a) (b) (c)

FIGURE 5: (a) The graphs to be matched. (b) The association graph that does not consider semantic labels. (c) The association graph that has considered semantic labels.

the weight matrix W to directly set the weights of conflicting matching pairs to zero during initialization, reducing the times of walking of the random walker. In addition, we consider the movement frequency of objects in the scene and believe that the objects with low movement frequency are more representative in images and are given greater weights when calculating image similarities.

(2) Node position difference: assume that dis_{pq} is the Euclidean distance between node p and node q. Obviously, the smaller dis_{ia} is, the higher the matching degree with v_i^M and v_a^N is; the smaller the difference between dis_{ij} and dis_{ab} is, the higher the matching degree with e_{ij}^M and e_{ab}^N is. According to [30], the weight matrix and probability transition matrix of nonconflicting matching pairs are initialized using the two following equations:

$$W_{ia;jb} = e^{-\left(\left(\text{dis}_{ij} - \text{dis}_{ab}\right)^2/\sigma_s^2\right)}, \tag{3}$$

$$\sigma_s^2 = 2500,$$

$$P = \frac{W}{\max ia \sum_{jb} W_{ia;jb}}. \tag{4}$$

(i) In equation (3), σ_s^2 is an adjustment factor.

(3) Topological structure difference: according to the preset path length and the times that each node in the association graph serves as the initial node, the random walk algorithm is executed on the graph. In this paper, the similarity between images can be expressed as

$$s = \frac{\sum_{i=1}^{q} m_i}{\sum_{i=1}^{q} m_i + \sum_{i=1}^{q} n_i + \sum_{i=1}^{q} V_i d}, \tag{5}$$

where s is the image similarity, m_i denotes the normalized weight of the i-th matching pair in graph G_{ass}, and n_i in the range of $(0, 1)$ represents the weight of the i-th unmatched object in the original images. The value of m_i is determined by the random walk algorithm, while n_i is determined by the movement frequency of the object in the scene. V_i denotes the pixel position deviation of the object corresponding to the i-th matching pair in the original images, and q is the number of matching pairs, while d is an adjustment factor determined by the number of objects in images and the acquisition frequency of the camera when constructing the map library; we take $d = 0.001$ in this paper. The algorithm for calculating the image similarity in this paper is given in Algorithm 1.

(i) **Inputs**: Semantic topological graphs of G^M and G^N to be matched, the path length of random walk: p, the times of each node used as a starting node: *count*
(ii) **Output**: Image similarity: s
(1) Construct the association graph: G_{ass}
(2) **for** i in c^M **do**
(3) **for** a in c^N **do**
(4) **if** v_i^M and v_a^N have the same semantic labels, **then**
(5) add v_{ia} to G_{ass}
(6) **end if**
(7) **end for**
(8) **end for**
(9) Initialize the weight matrix: W, set the weights of conflicting matching pairs to zero, the weights of non-conflicting matched pairs are $W_{ia;jb} = e^{-((\text{dis}_{ij} - \text{dis}_{ab})/\sigma_s^2)}$, $\sigma_s^2 = 2500$
(10) Initialize the probability transition matrix: $P = W/(\max ia \sum_{jb} W_{ia;jb})$
(11) **for** k in nodes of G_{ass} **do**
(12) Random Walk (G_{ass}, k, p, *count*)
(13) $X^{(t+1)T} = X^{(t)T} P$
(14) **end for**
(15) Get the best matching pairs
(16) $s = \sum_{i=1}^{q} m_i / (\sum_{i=1}^{q} m_i + \sum_{i=1}^{q} n_i + \sum_{i=1}^{q} V_i d)$

Algorithm 1: Image similarity calculation.

4.3. Relocalization. The visual relocalization of the UAV recovers its position and pose by obtaining the rotation matrix and translation vector of the UAV camera. Solving the position and pose of the camera corresponding to the motion of n 3D-to-2D points correspondences is the PnP (Perspective-n-Point) problem. EPnP is a noniterative solution to the PnP problem with a computational complexity of $O(n)$. Its key idea is to represent n 3D points as the weighted sum of 4 noncoplanar virtual control points and calculate the coordinate values of 4 control points in the camera coordinate system; then the position and pose of the camera can be obtained.

Assuming that the UAV camera is a small hole model and the internal parameters are known, the image currently captured by the UAV is G_1, which corresponds to image G_2 in the map library after scene matching. There are a total of n $(n \geq 4)$ semantic feature matching pairs, where the 2D positions of the semantic feature points in G_1 in the camera coordinate system are known, and the 3D positions of the semantic feature points in G_2 in the world coordinate system are known.

We let the 3D coordinates of the semantic feature points and the 4 virtual control points in the world coordinate system be given as

$$\begin{aligned} p_i^w &= [x_i^w, y_i^w, z_i^w], \quad i = 1, 2, \ldots, n, \\ c_i^w &= [x_i^{cw}, y_i^{cw}, z_i^{cw}], \quad i = 1, 2, 3, 4. \end{aligned} \tag{6}$$

The coordinates of semantic feature points in the world coordinate system can be expressed as the weighted sum of 4 control points' coordinates (see the following equation):

$$\begin{aligned} p_i^w &= \sum_{j=1}^{4} \alpha_{ij} c_j^w, \\ \sum_{j=1}^{4} \alpha_{ij} &= 1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{7}$$

Let c_j^c represent the coordinates of the j-th virtual control point in the camera coordinate system, R and t denote the camera external parameters, and the coordinates of the i-th semantic feature point in the camera coordinate system are p_i^c; then the relationships between them are as follows:

$$\begin{aligned} c_j^c &= [R|t] \begin{bmatrix} c_j^w \\ 1 \end{bmatrix}, \quad j = 1, 2, 3, 4, \\ p_i^c &= [R|t] \begin{bmatrix} p_i^w \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} \sum_{j=1}^{4} \alpha_{ij} c_j^w \\ \sum_{j=1}^{4} \alpha_{ij} \end{bmatrix} \\ &= \sum_{j=1}^{4} \alpha_{ij} [R|t] \begin{bmatrix} c_j^w \\ 1 \end{bmatrix} = \sum_{j=1}^{4} \alpha_{ij} c_j^c, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{8}$$

After obtaining the 3D coordinates in the camera coordinate system, the gravity of world coordinates and the camera coordinates of the semantic feature points is calculated to obtain matrix A, matrix B, and matrix H:

$$\begin{aligned} p_0^w &= \frac{1}{n} \sum_{i=1}^{n} p_i^w, \\ p_0^c &= \frac{1}{n} \sum_{i=1}^{n} p_i^c, \\ A &= \begin{bmatrix} p_1^{wT} - p_0^{wT} \\ \cdots \\ p_n^{wT} - p_0^{wT} \end{bmatrix}, \\ B &= \begin{bmatrix} p_1^{cT} - p_0^{cT} \\ \cdots \\ p_n^{cT} - p_0^{cT} \end{bmatrix}, \\ H &= B^T A. \end{aligned} \tag{9}$$

Performing singular value decomposition on H, the position R and pose t of the UAV can be recovered, where R is the rotation matrix containing the pose information, which can be interchanged with the three-axis rotation angle of the camera coordinate system, and t is a translation vector containing position information:

$$\begin{aligned} H &= U\sum V^T, \\ R &= UV^T = R_z(\varphi)R_y(\theta)R_x(\psi) \\ &= \begin{bmatrix} \cos\theta\cos\varphi & \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \cos\theta\sin\varphi & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \cos\psi\sin\theta\sin\varphi + \sin\psi\cos\theta \\ -\sin\theta & \sin\psi\cos\theta & \cos\psi\cos\theta \end{bmatrix}, \\ t &= p_0^c - rp_0^w = [X, Y, Z]^{\mathrm{T}}. \end{aligned} \tag{10}$$

5. Simulation

The experiments involved in this paper are all carried out on Ubuntu 18.04 system, using PyTorch to realize the YOLOv3 object detector. We use a UAV to shoot 800 pictures in an indoor scene at a rate of 10 fps to construct a scene map library and select 400 pictures from them as test set 1 and then shoot 50 pictures with different lighting conditions from the map library as test set 2. The precision and recall commonly used in machine learning are used as the evaluation indicators of our algorithm's performance. True positive (TP) means that the result of scene matching is correct, while false positive (FP) means that the result of scene matching is wrong, and false negative (FN) means that the relocalization of UAV fails; we have

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}. \end{aligned} \tag{11}$$

The precision and recall of the perfect relocalization system are both 100%; however, there is always a trade-off between them in the actual system. In the visual relocalization system, priority should be given to avoiding false positive and ensuring a sufficiently high precision rate, as result of introducing incorrect matching results during relocalization will lead to catastrophic failure. Aiming at the precision, recall, positioning error, and algorithm running time of the UAV relocalization algorithm, this paper compares the method we proposed with the visual relocalization method based on the bag-of-words model in test set 1 and test set 2.

5.1. Precision and Recall. To avoid the FP situation as much as possible, the image similarity threshold γ for judging the success of relocalization needs to be set, and its relationship with the relocalization result is written as

$$\text{relocalization result} = \begin{cases} \text{successed}, & \text{if } s > \gamma, \\ \text{failed}, & \text{if } s \le \gamma. \end{cases} \tag{12}$$

In test set 1, the precision and recall of the two visual relocalization methods with different similarity thresholds are shown in Figure 6. In the case of prioritizing the precision rate, the precision and recall are comprehensively considered to determine the best image similarity threshold γ for judging the success of relocalization. At this time, the precision and recall of the method we proposed are 99.23% and 97.23%, respectively, and the precision and recall based on the bag-of-words model are 100% and 99.25%, respectively.

Based on the best similarity thresholds $\gamma_1 = 0.01$ and $\gamma_2 = 0.425$, the two relocalization methods are compared in test set 1 and test set 2, and the results are shown in Table 1. When the lighting conditions of the UAV input images are the same as the map library, the visual relocalization effect based on the bag-of-words model is slightly better than the method proposed in this paper, and its precision and recall are 0.77% and 2.02% higher, respectively. Due to the need to prioritize higher precision in visual relocalization, the method we proposed also shows good performance. Generally speaking, the performance difference between the two methods is not obvious at this time. However, when the lighting conditions of the input images are different from those in the map library, the precision and recall achieved by the visual relocalization method based on the bag-of-words model drop sharply, reaching 40.74% and 32.35%, respectively, which are far from satisfactory for UAVs. This is because the changes in the gray distribution of the images caused by the changes in illumination have strengthened or weakened some of the key features, resulting in a large difference between the prediction of ORB feature points and the BoW descriptors compared with those of the original images, and the probability of occurrence of FP situation increases accordingly. On the other hand, because of the strong feature invariance of semantic information extracted by the object detection approach, the visual relocalization method proposed in this paper is far more capable of coping with changes in illumination than the visual relocalization method based on the bag-of-words model. The precision and recall are still 93.18% and 87.23%, respectively, and the precision rate and the recall rate are 52.44% and 54.88% higher than those of the visual relocalization method based on the bag-of-words model, respectively.

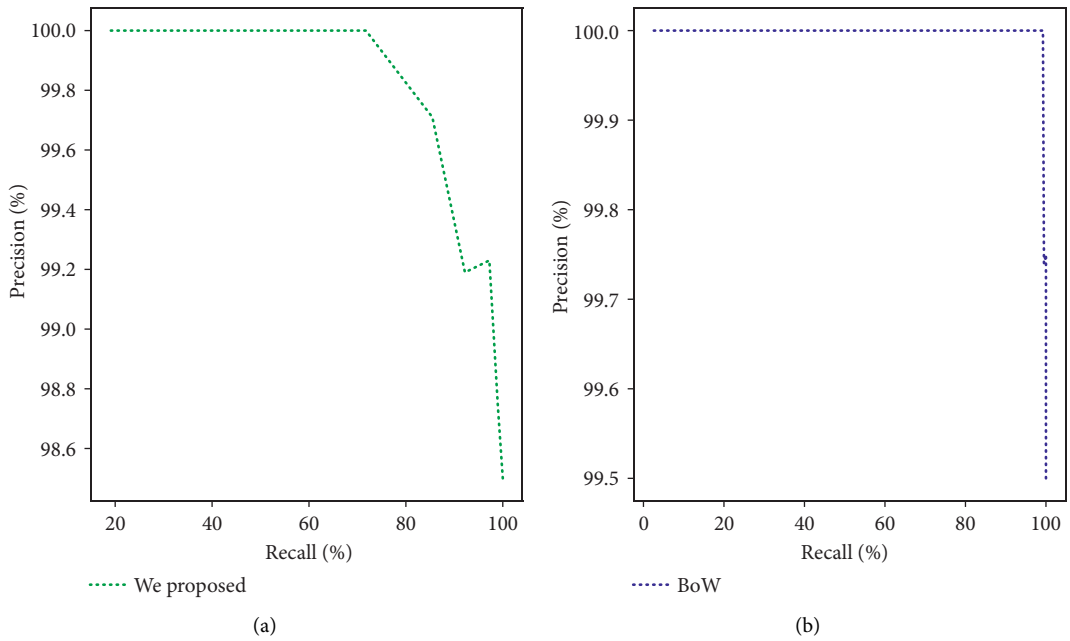


Figure 6: The precision and recall of the two visual relocalization methods with different similarity thresholds. (a) The method we proposed. (b) The BoW method using ORB features.

Table 1: Comparison of experimental results.

Method The test set	Proposed Test set 1	BoW Test set 1	Proposed Test set 2	BoW Test set 2
FP when no threshold is set	6	2	7	28
TP	386	397	41	11
FP	3	1	3	16
FN	11	2	6	23
Precision	99.23%	100%	93.18%	40.74%
Recall	97.23%	99.25%	87.23%	32.35%

5.2. Relocalization Error. Figures 7(a)–7(c) show the relocalization errors in test set 1, and Figures 7(d)–7(f) show the relocalization errors in test set 2. The peaks of the curves in Figure 7(a) and 7(b) correspond to the FP situation of scene matching. In addition, the X-axis, Y-axis, and Z-axis errors were obtained by the two relocalization methods in the UAV camera coordinate system control within ±10 cm. When the lighting conditions change, to intuitively reflect the distribution of the relocalization errors, we use a scatter chart to record the X-axis error, Y-axis error, and Z-axis error of the two relocation methods and use a frequency distribution table (see Table 2) to record the distance error frequency distribution of the two methods. According to Figures 7(d)–7(f), it can be seen that, due to the changes in the image appearance caused by the changes in illumination, a large number of FP situations appear in the relocalization results based on the bag-of-words model, and the overall distribution of the data points of the bag-of-words model is above that of the object semantic model we proposed, which means that the overall X-axis error, Y-axis error, and Z-axis error of the relocalization method based on the bag-of-words model are larger than those of the method we proposed.

It can be seen from Table 2 that, using the method proposed in this paper for visual relocalization, the proportion of samples with a distance error of less than 10 cm still accounts for 42%, and the proportion of samples with a distance error of less than 30 cm accounts for 86%, although the lighting conditions have changed. It is proved that the method we proposed is robust against changes in lighting conditions in the environment. While using the bag-of-words model for visual relocalization, the proportions of samples with a distance error of less than 10 cm and 30 cm are only 12% and 40%, respectively.

5.3. Running Time. We record the average running time required to complete a single scene matching for the two relocalization algorithms, respectively. The running time of the algorithm proposed in this paper is affected by the number of semantic features in the map library. The average number of semantic features contained in each image in the map library is set to 7. As shown in Table 3, the average running time required for the proposed algorithm to complete a single scene matching is 0.027 s faster than that of the bag-of-words model's algorithm known for its fast speed,

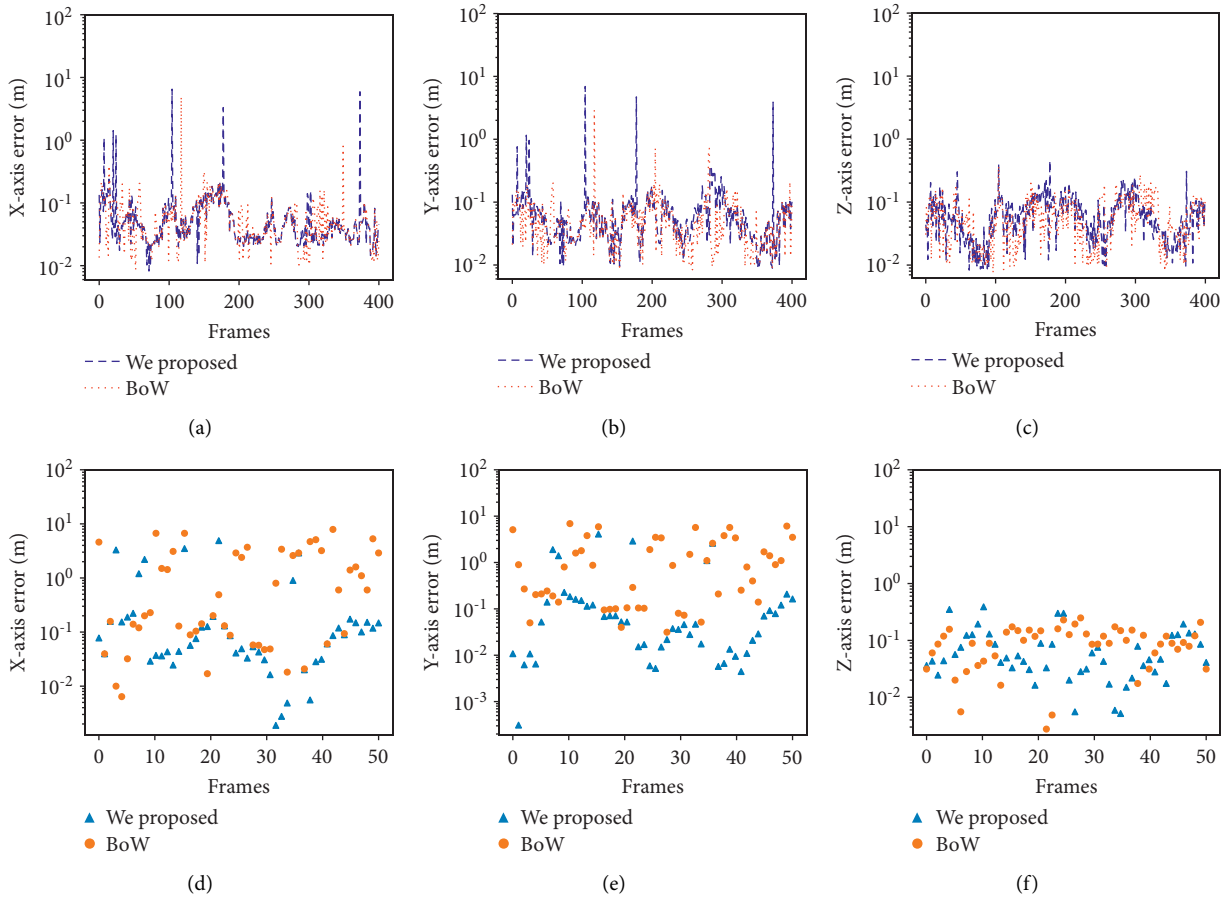


Figure 7: Relocalization error. (a) X-axis error of relocalization under constant lighting conditions. (b) Y-axis error of relocalization under constant lighting conditions. (c) Z-axis error of relocalization under constant lighting conditions. (d) X-axis error of relocalization under changing lighting conditions. (e) Y-axis error of relocalization under changing lighting conditions. (f) Z-axis error of relocalization under changing lighting conditions.

Table 2: Frequency distribution of relocalization distance error when illumination changes.

Relocalization method	Relocalization distance error (m)	Frequency	Proportion (%)
BoW	<0.1	21	42
	<0.3	43	86
Proposed	<0.1	6	12
	<0.3	20	40

Table 3: The average running time of the algorithm required to complete a single scene matching.

Scene matching algorithm	Proposed (s)	The BoW algorithm using ORB features (s)
Running time	0.619	0.646

which can meet the requirement of real-time visual relocalization of UAVs. Such a fast running speed benefits from the introduction of semantic features when constructing the association graphs. The process of removing conflict matching pairs with the help of semantic features is equivalent to pruning the association graphs, which reduces the times of randomly walking.

6. Conclusion

UAVs need to have accurate self-localization capabilities when performing IoT tasks; however, in the real world, complex lighting changes have brought huge challenges to the visual relocalization of UAVs. Oriented to the UAV-based IoT, this paper proposes a UAV visual relocalization

method using semantic object features to solve this problem. This method uses YOLOv3 as the object detection framework, extracts the semantic information in the images, and uses the semantic information to construct topological graphs as sparse descriptions of the environment. Then, with prior knowledge of the map, a random walk algorithm is used to perform semantic features matching as well as the scene matching. Finally, the EPnP algorithm is used to solve the UAV's position and pose, which will be returned to the IoT platform. After simulation experiments, the precision and recall of the method in this paper are only 0.77% and 2.02% lower than those of the visual relocalization method based on the bag-of-words model when the scene lighting conditions remain unchanged. Meanwhile the precision and recall are 52.44% and 54.88% higher than those of the visual relocalization method based on the bag-of-words model when the lighting conditions of the scene change dynamically, which proves the effectiveness and robustness of the method we proposed. The average running time of this method to complete a single scene matching is 0.027 s faster than that of the bag-of-words model's algorithm using ORB features, which can meet the requirement of UAVs' real-time visual relocalization and provide a guarantee for UAVs to perform IoT tasks.

Data Availability

The image data used to support the findings of this study have not been made available because they involve privacy.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 71731001, U1933130, 1433203, and U1533119) and was also supported by Zhejiang Key Laboratory of General Aviation Operation Technology (General Aviation Institute of Zhejiang JianDe) (no. JDGA2020-5) and Research Project of Chinese Academy of Sciences (no. ZDRW-KT-2020-21-2).

References

[1] X. Wang, Z. Ning, S. Guo, M. Wen, L. Guo, and V. Poor, "Dynamic UAV deployment for differentiated services: a multi-agent imitation learning based approach," *IEEE Transactions on Mobile Computing*, vol. 99, p. 1, 2021.

[2] C. M. Korpela, T. W. Danko, and P. Y. Oh, "Designing a system for mobile manipulation from an unmanned aerial Vehicle," in *Proceedings of the 2011 IEEE Conference on Technologies for Practical Robot Applications*, pp. 109–114, Woburn, MA, USA, April 2011.

[3] Z. Ning, P. Dong, M. Wen, X. Wang, and H. Vincent, "5G-Enabled UAV-to-community offloading: joint trajectory design and task scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, p. 99, 2021.

[4] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: a crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[5] J. Surber, L. Teixeira, and M. Chli, "Robust visual-inertial localization with weak GPS priors for repetitive UAV flights," in *Proceedings of the 2017 IEEE International Conference on Robotics & Automation*, pp. 6300–6306, Singapore, May 2017.

[6] Z. Ning, S. Sun, X. Wang et al., "Blockchain-enabled intelligent transportation systems: a distributed crowdsensing framework," *IEEE Transactions on Mobile Computing*, vol. 99, p. 1, 2021.

[7] B. Kakillioglu, J. Wang, S. Velipasalar, A. Janani, and E. Koch, "3D sensor-based UAV localization for bridge inspection," in *Proceedings of the 2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1926–1930, Pacific Grove, CA, USA, November 2019.

[8] A. Couturier and M. A. Akhloufi, "A review on absolute visual localization for UAV," *Robotics and Autonomous Systems*, vol. 135, Article ID 103666, 2021.

[9] Y. C. Liu and Q. H. Dai, "Vision aided unmanned aerial Vehicle autonomy: an overview," in *Proceedings of the 2010 3rd International Congress on Image and Signal Processing (CISP),*, vol. 1, pp. 417–421, Yantai, China, November 2010.

[10] X. Wang, Z. Ning, S. Guo, M. Wen, and V. Poor, "Minimizing the age-of-critical-information: an imitation learning-based scheduling approach under partial observations," *IEEE Transactions on Mobile Computing*, vol. 99, p. 1, 2021.

[11] Q. Liu, R. Liu, Z. Wang, and J. S. Thompson, "UAV swarm-enabled localization in isolated region: a rigidity-constrained deployment perspective," *IEEE Wireless Communication Letters*, vol. 99, p. 1, 2021.

[12] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Localization from semantic observations via the matrix permanent," *The International Journal of Robotics Research*, vol. 35, no. 1–3, pp. 73–99, 2016.

[13] W. Zhao, K. Qian, Z. Ma, X. Ma, and H. Yu, "Stereo visual slam using bag of point and line word pairs," in *Proceedings of the International Conference on Intelligent Robotics and Applications*, Springer, Shenyang, China, August 2019.

[14] D. Wang, H. Wang, K. Wang, and H. Mo, "Research on loop closing for slam based on RGB-D images," in *Proceedings of the 2018 Chinese Intelligent Systems Conference*, pp. 739–748, Wenzhou, China, January 2019.

[15] M. Cummins and P. Newman, "Fab-map: probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 47–65, 2008.

[16] S. Lowry, N. Sunderhauf, P. Newman et al., "Visual place recognition: a survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2015.

[17] M. Bansal, M. Kumar, and M. Kumar, "2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 39–57, 2021.

[18] Y. M. Jeon, I. H. Lee, K. B. Bae, H. G. Ji, and Y. S. Bae, "A study on the SIFT, SURF, and hog features of image in the field of face defect inspection," in *Proceedings of the Korean Society of Computer Information Conference*, January 2019.

[19] N. Malik, A. G. Airij, S. A. Memon, Y. N. Panhwar, S. A. R. Abu-Bakar, and M. A. El-Kh, "Performance comparison between SURF and SIFT for content-based image retrieval," in *Proceedings of the 2019 IEEE International Conference on Signal and Image Processing Applications*

(ICSIPA), pp. 214–218, Kuala Lumpur, Malaysia, September 2019.

[20] J. Huang, *Comparative Research on UAV Image Feature Matching Algorithm*, Geomatics & Spatial Information Technology, Heilongjiang, China, 2019.

[21] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," in *Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–10, Sukkur, Pakistan, March 2018.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, October 2017.

[23] S. Ren, K. He, R. Girshick, J. Sun, and R.-C. N. N. Faster, "Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 37–49, 2017.

[24] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," arXiv: arXiv:1804.02767, 2018.

[25] Y. Zhang, P. Zhao, D. Li, and K. Konstantin, "Spatial attention based real-time object detection network for Internet of Things devices," *IEEE Access*, vol. 8, pp. 165863–165871, 2020.

[26] Z. Bai and D. Jiang, "On the multi-scale real-time object detection using resnet," in *Proceedings of the Pattern Recognition and Computer Vision, Second Chinese Conference, PRCV 2019*, pp. 63–73, Xi'an, China, November 2019.

[27] J. Civera, D. Galvez-Lopez, L. Riazuelo, J. D. Tardos, and J. Montiel, "Towards semantic slam using a monocular camera," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems*, pp. 1277–1284, San Francisco, CA, USA, September 2011.

[28] N. Mandel, M. Milford, and F. Gonzalez, "Incorporating hierarchical information for UAV based semantic mapping," in *Proceedings of the 2021 IEEE Aerospace Conference (50100)*, pp. 1–11, Big Sky, MT, USA, March 2021.

[29] A. B. Salvado, R. Mendonca, A. Lourenco, F. Marques, and J. Barata, "Semantic navigation mapping from aerial multispectral imagery," in *Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pp. 1192–1197, Vancouver, BC, Canada, June 2019.

[30] A. G. Toudeshki, F. Shamshirdar, and R. Vaughan, "UAV visual teach and repeat using only semantic object features," in *Proceedings of the 2018 15th Conference on Computer and Robot Vision (CRV)*, pp. 182–189, Toronto, ON, Canada, May 2018.

[31] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *Proceedings of the Computer Vision–Eccv 2010*, pp. 492–505, Heraklion, Crete, Greece, September 2010.

[32] G. Li, J. Luo, Q. Xiao, C. Liang, P. Ding, and B. Cao, "Predicting microrna-disease associations using network topological similarity based on deepwalk," *IEEE Access*, vol. 5, pp. 24032–24039, 2017.

[33] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: an accurate O(N) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 55–66, 2018.

[34] Z. Ning, S. Sun, X. Wang et al., "Intelligent resource allocation in mobile blockchain for privacy and security transactions: a deep reinforcement learning based approach," *Science China Information Sciences*, vol. 64, no. 6, pp. 1–16, 2021.