

Article

Global Semantic Localization from Abstract Ellipse-Ellipsoid Model and Object-Level Instance Topology

Heng Wu, Yanjie Liu *, Chao Wang and Yanlong Wei

State Key Laboratory of Robotics and Systems (HIT), School of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China; 20b908039@stu.hit.edu.cn (H.W.)

* Correspondence: yjliu@hit.edu.cn

Abstract: Robust and highly accurate localization using a camera is a challenging task when appearance varies significantly. In indoor environments, changes in illumination and object occlusion can have a significant impact on visual localization. In this paper, we propose a visual localization method based on an ellipse-ellipsoid model, combined with object-level instance topology and alignment. First, we develop a CNN-based (Convolutional Neural Network) ellipse prediction network, DELLipse-Net, which integrates depth information with RGB data to estimate the projection of ellipsoids onto images. Second, we model environments using 3D (Three-dimensional) ellipsoids, instance topology, and ellipsoid descriptors. Finally, the detected ellipses are aligned with the ellipsoids in the environment through semantic object association, and 6-DoF (Degree of Freedom) pose estimation is performed using the ellipse-ellipsoid model. In the bounding box noise experiment, DELLipse-Net demonstrates higher robustness compared to other methods, achieving the highest prediction accuracy for 11 out of 23 objects in ellipse prediction. In the localization test with 15 pixels of noise, we achieve *ATE* (Absolute Translation Error) and *ARE* (Absolute Rotation Error) of 0.077 m and 2.70° in the *fr2_desk* sequence. Additionally, DELLipse-Net is lightweight and highly portable, with a model size of only 18.6 MB, and a single model can handle all objects. In the object-level instance topology and alignment experiment, our topology and alignment methods significantly enhance the global localization accuracy of the ellipse-ellipsoid model. In experiments involving lighting changes and occlusions, our method achieves more robust global localization compared to the classical bag-of-words based localization method and other ellipse-ellipsoid localization methods.

Keywords: semantic localization; instance topology; ellipse-ellipsoid model; illumination variations; object occlusion



Citation: Wu, H.; Liu, Y.; Wang, C.; Wei, Y. Global Semantic Localization from Abstract Ellipse-Ellipsoid Model and Object-Level Instance Topology. *Remote Sens.* **2024**, *16*, 4187. <https://doi.org/10.3390/rs16224187>

Academic Editor: Pablo Rodríguez-Gonzálvez and Baocheng Zhang

Received: 9 September 2024
Revised: 4 November 2024
Accepted: 8 November 2024
Published: 10 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robustly and accurately estimating a robot's 6-DoF pose in complex environments from a single image is a critical capability for visual SLAM (Simultaneous Localization and Mapping) and navigation [1–3]. However, seasonal changes, weather variations, differences between natural and artificial lighting, and continuous alterations in both static and dynamic objects can result in substantial discrepancies between the robot's observed appearance and the previously constructed map [4,5]. These discrepancies present significant challenges for robot localization tasks.

The localization component of traditional visual SLAM models the environment as a set of keyframe databases during the mapping process and matches the query image with keyframes through local feature extraction and comparison [6,7]. When the appearance difference between the query image and keyframes is small, it can achieve high localization accuracy. However, changes in illumination or object occlusion in the environment can significantly alter the appearance of images captured in the same scene, which may negatively affect the accuracy and robustness of localization [8,9]. Real-world environments contain many semantic objects whose 3D structure and semantic information remain time-invariant under certain conditions [10]. For example, a television in a room retains the

same structure and semantics regardless of time, and its spatial location tends to remain stable. Robots can recognize such objects in the environment using deep learning models like Faster R-CNN [11], or YOLO [12,13], which creates opportunities for robots to localize themselves, much like humans, by relying on these reliable semantic signposts. For instance, Li et al. [14] modeled objects as cuboids and used Faster R-CNN [11] to examine the observed bounding box of an object, aiming to construct a model for pose estimation based on the projection of the cuboid and the detected bounding box. However, the cuboid projection is not rectangular, requiring numerous assumptions for accurate pose estimation.

Abstracting objects in the environment as ellipsoids provides a more efficient solution. When a 3D ellipsoid is projected onto a 2D (Two-dimensional) image using the pinhole model, its center is projected to a pixel in the image. If the camera observes more than two ellipsoids, the center-pixel pairs form a classic PnP (Perspective-n-Point) problem. When the number of observed ellipsoids is exactly two, Vincent et al. [15] can approximately localize them based on their projections and rotation parameters. In the image, the projection of an ellipsoid appears as a quadratic curve (ellipse), and the pixel corresponding to the center of this curve represents the pixel projected from the ellipsoid's center. Therefore, the accuracy of ellipse prediction in 2D images is directly related to the overall localization precision. The above may only be achieved when ellipsoids and ellipses are accurately matched. YOLO et al. [12,13] can detect object bounding boxes and categories, where the bounding boxes can approximate the projections of ellipsoids, and the object categories can be matched with ellipsoids in the environment. However, in practical applications, noise in the bounding boxes and the presence of multiple objects from the same category may affect localization accuracy.

In this paper, we propose a pipeline based on ellipse-ellipsoid pairs for global localization. First, we designed an object-level instance topology and alignment method that accurately matches detected targets with objects. Second, we developed a robust ellipse prediction model, DEllipse-Net, which uses segmented RGB-D blocks to reliably estimate the projection of ellipsoids. Finally, we use the aligned ellipse-ellipsoid pairs to achieve precise global localization. Our main contributions are:

- We combine the semantic, 3D coordinates and topology information of objects to propose an object-level instance topology and alignment method, which can accurately align ellipses and ellipsoids and effectively improve the localization accuracy.
- We propose an ellipse prediction network (DEllipse Net) capable of accurately predicting the projections of 3D ellipsoids on images, even in the presence of bounding box noise. Additionally, this network is lightweight, easily portable, and can be adapted to multiple objects with a single model.
- We propose a global semantic localization system based on an ellipse-ellipsoid model and object-level instance topology, and evaluate its robustness to significant changes in appearance (light changes and object occlusion).

Our global localization system is shown in Figure 1 and contains the following parts. Object detection: Targets of interest in the query image are detected using YoloV8, and the RGB-D image are cropped according to the bounding box of the target. Ellipse prediction: Taking the cropped RGB-D image pairs as input, the projection (ellipse) of the object is estimated using DEllipse-Net. Object-level instance topology: The objects (ellipses) detected in the query image are subjected to 3D instance topology and the descriptors of each ellipse are extracted. Global map: Consists of a set of ellipsoids with category, size, and 3D coordinates representing different objects in the environment. The ellipsoids are associated through an object-level instance topology. Pose estimation: Position estimation based on matched ellipse-ellipsoid pairs.

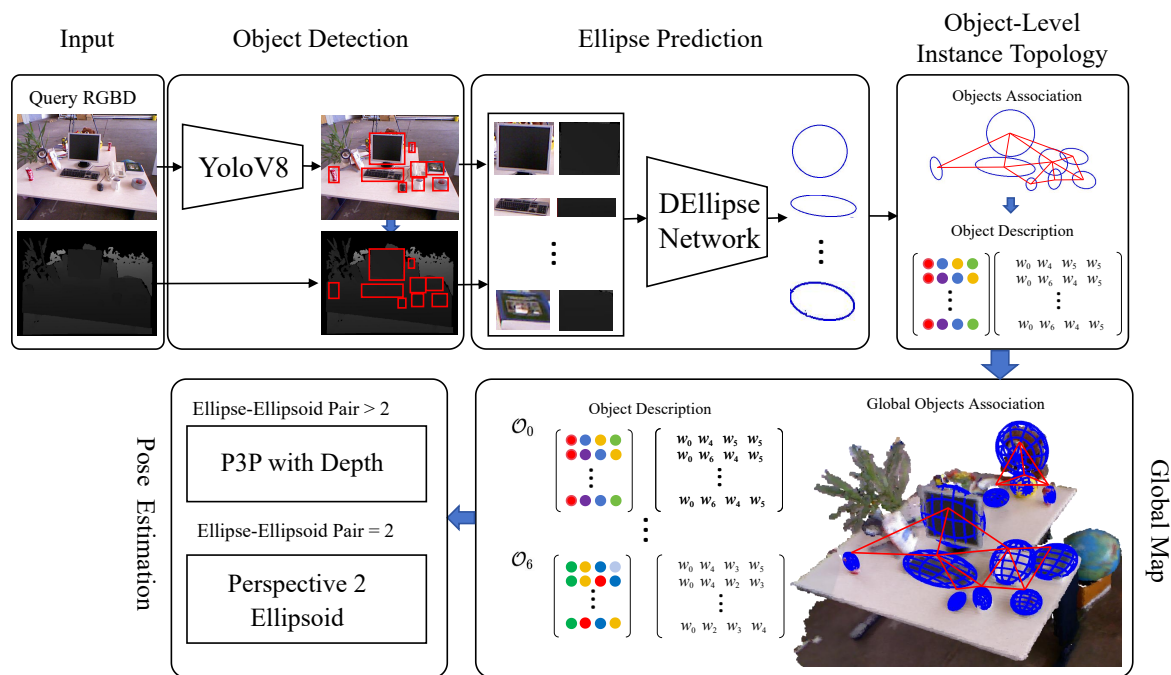


Figure 1. Overview of global semantic localization. Before using a single-frame RGB-D image for localization, it is essential to construct a “Global Map” that includes the 3D ellipsoid abstraction and the topological relationships of each object instance. Upon receiving each RGB-D image as input, we use YOLOv8 [13] to detect the semantic information and bounding boxes of each object within the RGB image. We then crop the corresponding object regions from both the RGB and depth images for ellipse prediction using DEllipse-Net. Afterward, the ellipse and semantic information are combined to construct a topology map, which is aligned with the “Global Map.” Finally, the aligned ellipse-ellipsoid pairs are used for pose estimation.

2. Related Work

Visual global localization is a widespread concern in visual SLAM. Based on the appearance, we categorize them into feature-based and semantic-based.

2.1. Global Localization Based on Feature Point

Global localization using BoW (bag-of-words) [16] method with points as features is the most common approach. FAB-MAP [17] uses BoW [16] and Chow-Liu trees [18] to infer the correspondence between the feature points in the query image and the 3D map points, and to perform camera pose estimation. DBoW2 [19] constructs an image database based on inverse-order file structure using ORB (Oriented FAST and Rotated BRIEF), SIFT (Scale Invariant Feature Transform), etc. as descriptors for indexing images and fast querying. Visual SLAMs such as ORB-SLAM [20–22] and Vins-mono [23] use DBoW2 for relocation and loop-closing. However, this approach relies on feature matching and is not robust enough when there are strong changes in appearance.

To better cope with changes in appearance, recent research has turned to the use of more advanced visual features. HF-Net [24] proposes a hierarchical localization method based on CNN to predict local and global features. Following a coarse-to-fine localization paradigm, global retrieval is first performed to obtain a set of database images, and then these maps are clustered into positions using the covariance of the 3D-SfM (Structure from Motion) model, and 2D-3D matching of local features at these candidate positions is performed to obtain the camera’s accurate 6-DoF position. LoFTR [25] proposes a new method for local image feature matching, using the global receptive field provided by Transformer [26] makes it possible to produce dense matches in low texture regions based on the feature descriptors of the two images. The global receptive field provided by the Transformer allows LoFTR to produce dense matches in less textured regions where it is

often difficult for the feature detector to produce usable points of interest. VS-Net [27] proposes a voting network that uses segmentation to partition pixels into distinct landmark blocks and uses landmark location voting branches to estimate landmark locations within each block. The framework establishes a correspondence between a query image and a specific scene with a learnable set of landmarks by landmark matching to predict accurate bit positions.

2.2. Global Localization Based on Semantic Object

Semantic feature based localization is mainly dependent on the development of image detection techniques. [28] utilizes object dense semantics and environment understanding for global localization. The environment with three-bit dense semantics, semantic graph and its topology is first modeled. Then, semantic objects are associated using random walks to use this object-level representation for position recognition, followed by 6-DoF pose estimation via semantic-level point alignment. [29] modeled object-level features in the environment using voxels and rectangles to represent the environment as a semantic graph with topological information. Based on this, an effective edit-distance based graph matching method is proposed to realize the alignment of the detection target and the objects in the environment for cyclic correction. Although this method has good localization ability and robustness to the environment, it is not able to localize accurately from a single image.

Recent work has proposed a solution for global localization using ellipse-ellipsoid pairs from a single image. Vincent et al. [30] introduced a method to infer camera translation from a pair of ellipsoids when the rotation matrix is known, addressing the issue of failed repositioning under significant appearance changes. This method leverages inertial sensors to provide necessary camera rotation parameters. Later, Vincent et al. improved on [15] by developing a method to restore the full pose through the correspondence between two ellipses. Refs. [15,30] involves constructing error functions based on the projection relationship between 2D ellipses and 3D ellipsoids. Consequently, accurate ellipse prediction becomes crucial for positioning accuracy. In both [15,30], the inscribed ellipse of the YOLO's bounding box is directly used as the ellipse projection. However, since the bounding box is axis-aligned and may contain errors in both size and position, this reduces localization accuracy. To address this, Zins et al. [31] developed an ellipse estimation network that infers ellipses from the RGB image within the bounding box, overcoming the limitations of directly fitting ellipses to axis-aligned bounding boxes, and demonstrating robustness to changes in bounding boxes. Nevertheless, to ensure accurate ellipse predictions, Zins et al. [31] constructed a separate network model for each object in the environment, which is clearly impractical. Moreover, the above methods did not accurately match ellipses to ellipsoid pairs in the early stages of pose estimation, instead relying on rough IoU-based (Intersection over Union) judgments between the projected and detected ellipses. This approach introduces unnecessary errors when there are multiple objects of the same category present in the environment.

Therefore, to enhance the localization accuracy of ellipsoids, we first designed an object-level topology and alignment method to accurately match detected ellipses with the corresponding ellipsoids in the environment (Section 3). Next, we introduced depth information into the ellipse prediction network to improve target-background separation, thereby enhancing the network's robustness and accuracy in predicting ellipses (Section 4). Finally, we combine these two approaches to achieve precise and robust global localization.

3. Global Ellipsoid Mapping and Object-Level Instance Topology and Alignment

When applying the ellipsoid model for global localization, matching detected ellipses with their corresponding environmental ellipsoids is critical. Ideally, recognized targets in the image could be matched directly to environmental objects based on their categories. However, it is common for multiple objects of the same category to coexist in the environment. To address this, we not only establish an abstract ellipsoid map for localization,

but also design object-level instance topology and alignment methods. These approaches associate environmental objects with image targets, enabling accurate matching.

3.1. Global Ellipsoid Mapping Construction

To solve the global localization problem using the ellipse-ellipsoid model, a global ellipsoid map is required. In this work, the ellipsoid map is constructed once and remains static to support long-term localization. We use ORB-SLAM3 [22] to compute the camera pose and generate a point cloud from the RGB-D data of the keyframes through camera interpolation. Voxelblox [32] then uses the point cloud and keyframe poses as input, constructing a dense map using its “simple” mode.

Figure 2a shows a dense map constructed with ORB-SLAM3 [22] and Voxelblox [32]. We select objects of interest in the map and segment them. As shown in Figure 2b, we use a suitable bounding box to enclose each object. In Figure 2c, the segmented objects are assigned corresponding semantic information. The bounding box simulates the size, location, and rotation of each object in the environment. Then, we represent each object using the outer ellipsoid of the bounding box and associate it with its semantic label. The ellipsoid $\mathcal{O}$ is expressed as:

$$4\gamma \cdot \left(\frac{x^2}{W^2} + \frac{y^2}{H^2} + \frac{z^2}{D^2} \right) = 1 \quad (1)$$

where, $[W, H, D]$ are the three axes of the bounding box. The center and rotation of the ellipsoid are consistent with the bounding box. $\gamma = 0.64$ is used to control the size of the ellipsoid to ensure that it can completely wrap around the object.

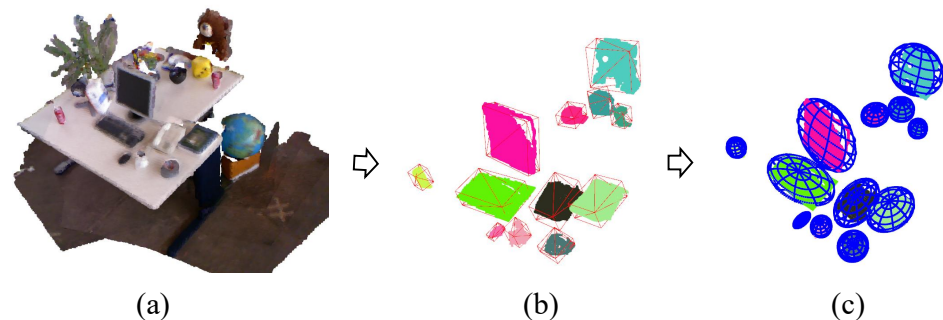


Figure 2. The construction process of ellipsoid map. (a) is the dense map constructed by ORB-SLAM3 and Voxelblox. (b) is the semantic map of objects. (c) is the ellipsoid map of objects.

3.2. Object-Level Instance Topology

When multiple objects of the same type exist in the environment, matching them by category is challenging. Humans often distinguish these objects based on their spatial relationships and surrounding context. Therefore, it is essential not only to use ellipsoids to represent the size and position of objects in the environment, but also to correlate them with nearby objects.

We represent the ellipsoid map with semantic labels introduced in Section 3.1 as nodes, using the ellipsoid’s center as the node’s coordinates in the world coordinate system. As shown in Figure 3, nodes of different colors are used to represent ellipsoids with different semantics, and similar objects may appear many times in the environment. Nodes are connected by undirected edges based on Euclidean distance, forming a topological graph that incorporates both distance and semantic information. To describe the distance relationships between node $\mathcal{O}_i$ and other nodes, the environment is uniformly divided into multiple spheres with radius $r_i = r * i$ centered at $\mathcal{O}_i$. r is a fixed radius, and r_i is the radius of the i -th sphere. If the distance d between $\mathcal{O}_{i+1}$ and $\mathcal{O}_i$ falls within $[r_i, r_{i+1}]$, the distance weight of $\mathcal{O}_{i+1}$ relative to $\mathcal{O}_i$ is $W_i = e^{-r_i}$. By centering at different nodes, we calculate the distance weights between each central node and the surrounding nodes. We assume

that the closer other nodes are to $\mathcal{O}_i$, the stronger their intrinsic connections, and thus, the greater the distance weight.

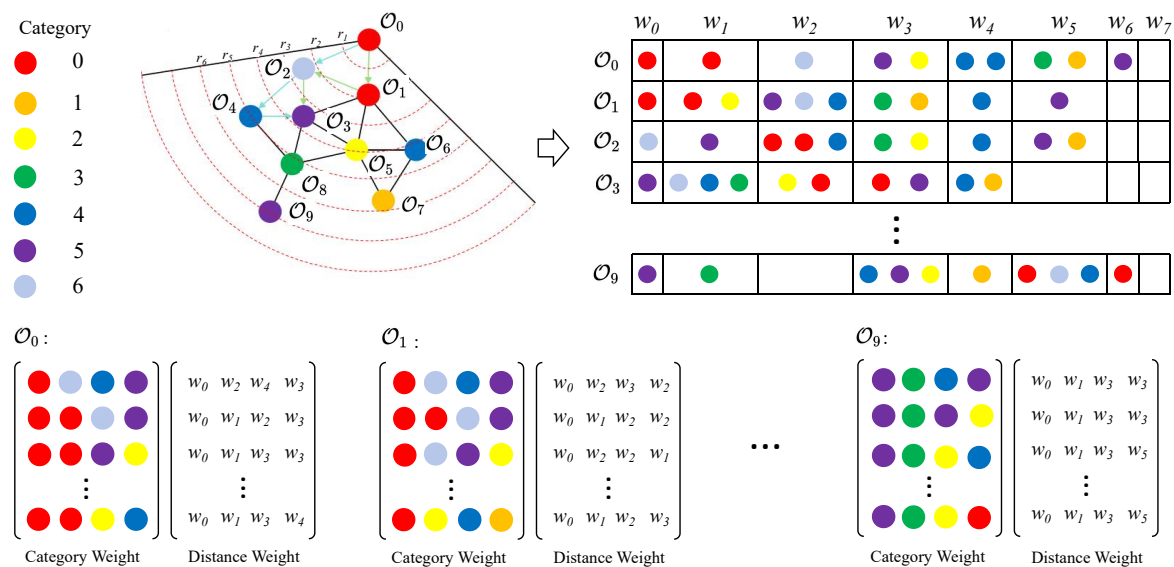


Figure 3. Object-level instance topology and descriptor extraction. Top left: undirected topology diagram between nodes and distance weights and random walks diagram with $\mathcal{O}_0$ as the root node. Top right: List of distance weights for each node. Bottom: Category and distance weights obtained through random walks. Nodes of different colors represent objects of different categories.

Then take $\mathcal{O}_i$ as the root node of the random walk, record the category and distance weight of the nodes passed by the random walk, and use it as a descriptor for the node in the environment. Ellipsoids in the environment are represented as $\{S_k^o\}^M$, M is the number of ellipsoids, k is the ID of ellipsoids. Descriptor of each object $S_k^o = (\mathbf{C}_k^o, \mathbf{W}_k^o)$, where $\mathbf{C}_k^o$ is the category matrix, and $\mathbf{W}_k^o$ is the weight matrix. $\mathbf{C}_k^o$ and $\mathbf{W}_k^o$ are both $T \times D$ matrices, T is the step size of random walk and D is the depth of random walk. T and D are set based on the number of ellipsoids in the environment. In this paper, $T = 4$ and $D = 50$.

As shown in Figure 3, starting from the root node $\mathcal{O}_0$, we perform a random walk with a step size of 4 in the topology graph. In the first step, we record the category 0 and weight w_0 of the root node. In the second step, the walk moves to $\mathcal{O}_2$, and we record its corresponding category 6 and weight w_2 . In the third step, the walk proceeds to $\mathcal{O}_4$, where we record category 4 and weight w_4 . In the fourth step, the walk continues to $\mathcal{O}_3$, recording category 5 and weight w_3 . At this point, we have completed one random walk with a step size of 4. We then repeat the above steps, starting a new walk from the root node and recording the category and weight labels at each step. After completing D walks, we have thoroughly recorded the category and weight descriptors $\mathcal{S}_k^o = (\mathbf{C}_k^o, \mathbf{W}_k^o)$ for $\mathcal{O}_0$ within the environment.

3.3. Descriptor Extraction of Query Image and Object Association of Environment

For querying images, we first use predicted ellipses to crop the depth image and then project the cropped image into point cloud segments. The centroid of each point cloud segment is used as the center of the corresponding object. We then apply the method described in Section 3.2 to associate the objects and extract their descriptors. The object in the query image is represented as $\{S_k^q\}^N$, where $S_k^q = (\mathbf{C}_k^q, \mathbf{W}_k^q)$, with N being the number of detected ellipses and k is the ID of each detected ellipse.

When ellipses and ellipsoids belong to different categories, they do not match. If there are multiple ellipsoids of the same type in the environment, we use similarity scores to

distinguish them. For query ellipse $\mathcal{S}_k^q$ and ellipsoid in the global map $\mathcal{S}_k^o$, the similarity is calculated as:

$$\text{Score}(\mathcal{S}_k^q, \mathcal{S}_k^o) = \frac{\sum_{i=1}^D \sum_{j=1}^D C_{ij} \cdot W_{ij}^T}{D^2} \quad (2)$$

$$C_{ij}[n] = \begin{cases} 1 & \text{if } \mathbf{C}_i^q[n] = \mathbf{C}_j^o[n] \\ 0 & \text{if } \mathbf{C}_i^q[n] \neq \mathbf{C}_j^o[n] \end{cases} \quad (3)$$

$$W_{ij}[n] = e^{-|\mathbf{W}_i^q[n] - \mathbf{W}_j^o[n]|} \quad (4)$$

where $\mathbf{C}_i^o$ is the category vector in the i -th row of $\mathbf{C}_k^o$, $\mathbf{C}_j^q$ is the category vector in the j -th row of $\mathbf{C}_k^q$, n denotes the position of elements in the vector. C_{ij} is the AND operation between $\mathbf{C}_i^o$ and $\mathbf{C}_j^q$ on the elements, used for filtering descriptions of the same category. $\mathbf{W}_i^o$ is the distances vector in the i -th row of $\mathbf{W}_k^o$, $\mathbf{W}_j^q$ is the distances vector in the j -th row of $\mathbf{W}_k^q$. W_{ij} is used to calculate the difference in distance at the same position. The higher the score, the higher the similarity.

4. Ellipse Prediction Network and Global Pose Estimation

The key to ellipse-ellipsoid localization lies in predicting the projection of a 3D ellipsoid onto the image, as accurate ellipse prediction can significantly improve localization accuracy. In this section, we propose an ellipse prediction method that combines depth information and employs the methods of Vincent et al. [15] and Zins et al. [31] for global localization.

4.1. Ellipse Estimation Network

Ellipse estimation is a method for predicting the projection of a 3D ellipsoid in an image, which relies on the bounding box from target detection. A approach is to use the inscribed ellipse of the bounding box as the projection of the ellipsoid. However, since the bounding box is aligned with the coordinate axes and contains noise, this can result in errors in the predicted ellipse. Another approach leverages the powerful capability of neural networks to extract intrinsic information from images, accurately predicting ellipses based on differences between the target and the background. In the real world, most objects exhibit geometric structures with convex surfaces. Based on the assumption of convex bodies, targets and backgrounds in the environment can be segmented. Depth information provides the geometric structure of objects, and using it can help distinguish objects within the environment. Therefore, we combine RGB and depth information to construct an ellipse prediction model.

The neural network part of the system is shown in Figure 1, consisting of YOLOv8 [13] for object detection and our ellipse prediction network. First, the camera's intrinsic parameters are used to calculate the 3D spatial coordinates corresponding to each pixel in the depth image, generating a point cloud of dimensions $W \times H \times 3$. Next, the bounding boxes provided by YOLOv8 are used to crop both the RGB image and the point cloud. Our ellipse prediction network, DEllipse-Net, takes the target's RGB image and a square subset of the point cloud as inputs, and resizes them to 256×256 using interpolation.

Our network, shown in Figure 4, consists of a base convolutional network for feature extraction, followed by MLP (Multi-layer Perceptron), several fully connected layers, and finally three parallel branches predict the ellipse parameters: center, axis and angle. The sizes of the center and axis are normalized relative to the input data dimensions 256×256 . To achieve this, the *Sigmoid* activation function is applied to constrain both values within the range $(0, 1)$. When calculating the difference between the predicted ellipse and the actual ellipsoid projection, the center and axes are scaled back to their original image dimensions using $(W_{bb}, H_{bb})/256$. Where W_{bb} and H_{bb} are the width and height of the bounding box. We define the right half of the ellipse as the direction of the ellipse, which can be fully described by the sine value. In the network, we use the *Tanh* function to restrict

the direction value to the range $(-1, 1)$, with the output representing the positive rotation of the angle. The detailed network structure is shown in Table 1.

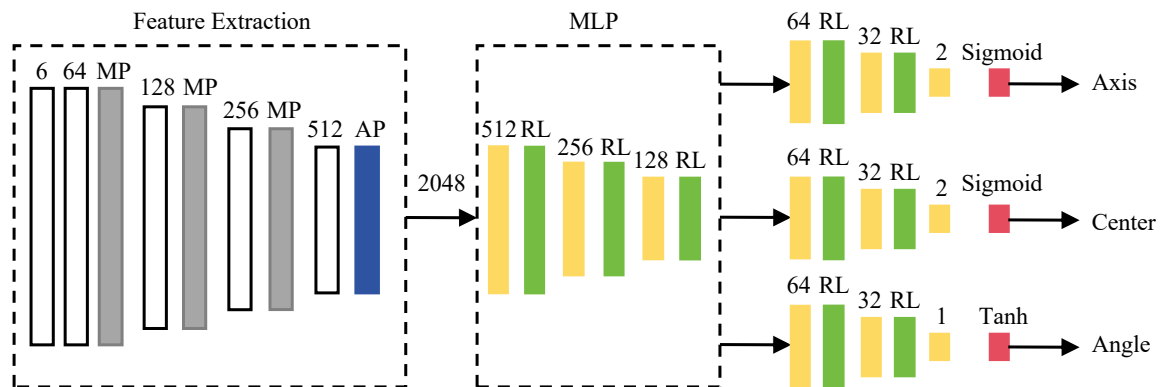


Figure 4. Overview of the DEllipse-Net. “MP” is maximum pooling, “AP” is average pooling, “RL” is Relu.

Table 1. Detailed network parameters of DEllipse-Net.

	Lary	Input	Output	Size	Stride	Padding	Activation
Feature	Conv2d	$6 \times 256 \times 256$	$64 \times 256 \times 256$	3	1	1	Relu
	Conv2d	$64 \times 256 \times 256$	$64 \times 256 \times 256$	3	1	1	Relu
	MaxPool2d	$64 \times 256 \times 256$	$64 \times 256 \times 256$	2	2	0	-
	Conv2d	$64 \times 256 \times 256$	$128 \times 256 \times 256$	3	1	1	Relu
	MaxPool2d	$128 \times 256 \times 256$	$128 \times 256 \times 256$	2	2	0	-
	Conv2d	$128 \times 256 \times 256$	$256 \times 256 \times 256$	3	1	1	Relu
	MaxPool2d	$256 \times 256 \times 256$	$256 \times 256 \times 256$	2	2	0	-
	Conv2d	$256 \times 256 \times 256$	$512 \times 256 \times 256$	3	1	1	Relu
	MaxPool2d	$512 \times 256 \times 256$	$512 \times 256 \times 256$	2	2	0	-
	Conv2d	$512 \times 256 \times 256$	$512 \times 256 \times 256$	3	1	1	Relu
	AdaptiveAvgPool2d	$512 \times 256 \times 256$	$512 \times 2 \times 2$	-	-	-	-
MLP	Linear	2048	256	-	-	-	-
	BatchNorm1d	256	256	-	-	-	-
	Relu	256	256	-	-	-	-
	Linear	256	256	-	-	-	-
	BatchNorm1d	256	256	-	-	-	-
	Relu	256	256	-	-	-	-
	Linear	256	64	-	-	-	-
	BatchNorm1d	64	64	-	-	-	-
	Relu	64	64	-	-	-	-
Axis	Linear	64	32	-	-	-	-
	BatchNorm1d	32	32	-	-	-	-
	Relu	32	32	-	-	-	-
	Linear	32	2	-	-	-	-
	Sigmoid	2	2	-	-	-	-
Center	Linear	64	32	-	-	-	-
	BatchNorm1d	32	32	-	-	-	-
	Relu	32	32	-	-	-	-
	Linear	32	2	-	-	-	-
	Sigmoid	2	2	-	-	-	-
Angle	Linear	64	32	-	-	-	-
	BatchNorm1d	32	32	-	-	-	-
	Relu	32	32	-	-	-	-
	Linear	32	1	-	-	-	-
	Tanh	1	1	-	-	-	-

4.2. Loss Function

In Section 4.1, we obtained three parameters representing ellipses: center, axis, and angle. Based on these three parameters, the expression for an ellipse is

$$\Psi(x) = (x - c)^T \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \frac{1}{a^2} & 0 \\ 0 & \frac{1}{b^2} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T (x - c). \quad (5)$$

where, x is the point on the ellipse, c is the center coordinate, a and b are the two axes of the ellipse, θ is the angle. But, this representation may not effectively integrate the axis and position components with the azimuth angle due to the discontinuity of the directional angle. Even a slight rotation will lead to significant differences in ellipses that share the same axis and center. To mitigate this effect, we adopt the new ellipse representation proposed in [31]:

$$\Psi(x) = (x - c)^T \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^T (x - c), \quad (6)$$

for ellipse representation. This representation of the ellipse allows for a natural handling of the discontinuity in the angle parameter.

To measure the difference between the predicted ellipse and the true ellipse, we discretize the continuous ellipse as follows:

$$\hat{\Psi}(x) = \begin{cases} D(x) & \text{if } x \text{ inside } \Psi(x) \\ -D(x) & \text{if } x \text{ outside } \Psi(x) \\ 0 & \text{if } x \in \Psi(x) \end{cases} \quad (7)$$

where, $D(x)$ is the distance from coordinate x to the ellipse curve. A 25×25 sampling grid is then employed to sample the ellipse. The loss of the network is

$$Loss = \sum_{i=1}^N (\hat{\Psi}_{pred}(p_i) - \hat{\Psi}_{gt}(p_i)), \quad (8)$$

where, p_i is the coordinate of the sampling point.

4.3. Pose Estimation from Ellipse to Ellipsoid

The 2D/3D object association list serves as the input for pose estimation. In Section 3.3, we match the ellipse in the query image with the ellipsoid in the environment to obtain a set of corresponding ellipse and ellipsoid pairs.

$$\Psi_k^q : \{\mathcal{O}_i^o, \mathcal{O}_j^o, \dots\} \quad (9)$$

Ψ_k^q denotes the ellipse in the image, while $\mathcal{O}_k^o$ represents the ellipsoids in the environment that belong to the same category as Ψ_k^q , listed in descending order according to their scores.

When there are three or more ellipse-ellipsoid pairs in the environment, the centers of the ellipses and the ellipsoids form corresponding point pairs. At this stage, we employ the P3P (Perspective-3-Point) method to estimate the pose and calculate the projection of each ellipsoid onto the image, as well as the Intersection over Union (IoU) of the ellipse under this pose. We use the IoU to select the optimal pose solution. Unlike [31], which uses an elliptic cone to infer the center, we directly utilize the center of the estimated ellipse and assign an initial depth from a depth image to accelerate convergence.

When there are two ellipse-ellipsoid pairs, we utilize the method from [15] for pose estimation. Assuming that the camera roll is negligible, this method simplifies the 6-DoF problem into a more manageable problem with only one remaining degree of freedom, corresponding to an angular parameter.

5. Experimental and Results

In this section, we evaluate our method using a publicly available RGB-D dataset. Our approach, which incorporates an ellipse-ellipsoid model and object topology alignment, demonstrates robust global localization performance, even under challenging conditions such as lighting variations and object occlusions. All experiments were conducted on a computer equipped with an Intel Core i5-13400K CPU running at 2.5 GHz and a GeForce RTX 4060 Ti GPU. The network was implemented using the PyTorch framework <https://pytorch.org/>.

5.1. Ellipse Prediction

We used the TUM dataset [33] (the *fr2_desk* sequence) and the 7-Scenes dataset [34] (the *Chess* sequence) to evaluate DEllipse-Net for ellipse prediction. In the *fr2_desk* sequence, we selected 11 categories and 12 objects to construct the ellipsoid environment. A total of 1000 images were selected for training DEllipse-Net and YOLOv8, while 500 images were reserved for testing. In the *Chess* sequence, we selected 7 categories and 11 objects to construct the ellipsoid environment. Six sequences totaling 6000 images were randomly selected, with 3000 images used for training and the remaining 3000 images for testing. During network training, the batch size was set to 16 for 200 epochs, using the Adam optimizer with an initial learning rate of 5×10^{-5} .

To demonstrate the superiority of our proposed ellipse network, we compared it with two other methods: one based on a classical mathematical ellipse (Mathematical) [15], and the other, an ellipse prediction network proposed by Zins et al. (Ellipse-Net) [31]. We evaluated the performance using four metrics: *IoU* (Intersection over Union), *IoU_{avg}*, *IoU_{Error}*, *ATE* (Absolute Translation Error), and *ARE* (Absolute Rotation Error). *IoU* measures the overlap between the predicted ellipse and the ground truth ellipse, with higher values indicating closer alignment between the predicted and actual ellipsoid projection. *IoU_{avg}* is the average *IoU* of all predicted objects of the same category. *IoU_{Error}* represents the difference between the predicted *IoU* and the ideal *IoU*, with smaller errors indicating better accuracy in predicting the ellipsoid projection. *ATE* assesses the translation error, with lower values indicating that the estimated translation is closer to the true translation. *ARE* measures the rotational error, with lower values reflecting smaller discrepancies between the predicted and ground truth rotation.

$$IoU = \frac{\text{Ell}_{gt} \cap \text{Ell}_{pre}}{\text{Ell}_{gt} \cup \text{Ell}_{pre}} \quad (10)$$

$$IoU_{avg} = \frac{\sum_{i=1}^M IoU_i}{M} \quad (11)$$

$$IoU_{error} = 1 - \frac{\text{Ell}_{gt} \cap \text{Ell}_{pre}}{\text{Ell}_{gt} \cup \text{Ell}_{pre}} \quad (12)$$

$$ATE = \left(\frac{1}{N} \sum_{i=1}^N \|\mathbf{T}_{pre} - \mathbf{T}_{GT}\|^2 \right)^{\frac{1}{2}} \quad (13)$$

$$ARE = \left(\frac{1}{N} \sum_{i=1}^N \left\| \frac{180}{\pi} \cos^{-1} \left(\frac{\text{trace}(\mathbf{R}_{gt}^{-1} \mathbf{R}_{pre} - 1)}{2} \right) \right\|^2 \right)^{\frac{1}{2}} \quad (14)$$

where, Ell_{pre} denotes the predicted ellipse, and Ell_{gt} represents the ground truth ellipse projected from the ellipsoid. $\mathbf{T}_{pre}$ and $\mathbf{T}_{gt}$ are the estimated translation vector and the true translation vector, respectively. $\mathbf{R}_{pre}$ and $\mathbf{R}_{gt}$ denote the estimated rotation matrix and the true rotation matrix, respectively.

We applied noise of 5 pixels, 10 pixels, and 15 pixels to the bottom left and top right vertices of the bounding box to evaluate the robustness of the three methods for ellipse prediction. Table 2 presents the *IoU_{avg}* variation of ellipse predictions and corresponding

ellipsoid projections for different objects under bounding box noise across two datasets. The table shows that when the detection box is undisturbed, the IoU_{avg} values of all three methods achieve good results of 0.9 or above. However, as the noise increases, the IoU_{avg} values for the three methods decrease. Notably, among the 23 objects in the two datasets, our method did not attain the highest IoU_{avg} value for 5 of these objects. Additionally, our method did not achieve the highest IoU_{avg} value in the prediction of 7 objects under partial disturbance. Nevertheless, for the remaining 15 objects, our method produced the best results. This indicates that our approach effectively addresses the inaccuracies in ellipse prediction caused by boundary detection errors. Figure 5 illustrates the performance of the three methods across the two datasets with a noise of 15 pixels. The white ellipse represents the true ellipsoid projection, while the white bounding box indicates the area with added noise. All three methods of ellipse prediction were performed within this region. It is evident that the added disturbance has the greatest impact on the Math method, while Ellipse-Net demonstrates some degree of robustness against interference. Our method, however, maintains relatively strong performance even after noise, for instance, in the *fr2_desk* dataset, our predictions almost completely overlap with the true ellipsoid.

With the enhancement of the ellipsoid prediction's anti-interference ability, there is a corresponding improvement in localization capability. Table 3 illustrates the impact of three ellipsoid prediction methods on localization performance under various noise. It is important to note that our testing was conducted using object-level topology and alignment, with the variables in this test limited to the three different ellipsoid prediction methods. The table demonstrates that, under different disturbances, the stable ellipsoid predictions generated by DEllipse-Net significantly enhance the robustness of ellipsoid-based localization. In eight localization experiments, we achieved the best results in seven of them.

Table 2. The IoU_{avg} values of each objects in *Chess* and *fr2_desk* under different bounding box noise. “Red”, “Blue”, “Purple”, and “Orange” represent the best values under the noise of 0 pixel, 5 pixel, 10 pixel, and 15 pixel noise, respectively.

Dataset	Object	Mathematical				Ellipse-Net				Ours			
		0 pix	5 pix	10 pix	15 pix	0 pix	5 pix	10 pix	15 pix	0 pix	5 pix	10 pix	15 pix
<i>fr2_desk</i>	TV	0.960	0.854	0.821	0.787	0.965	0.903	0.887	0.868	0.979	0.912	0.896	0.878
	Book	0.932	0.746	0.696	0.657	0.943	0.850	0.822	0.796	0.958	0.856	0.827	0.805
	Mouse	0.813	0.540	0.469	0.404	0.902	0.701	0.634	0.558	0.896	0.660	0.592	0.518
	Keyboard	0.785	0.710	0.682	0.647	0.936	0.892	0.878	0.862	0.930	0.883	0.865	0.840
	Cup	0.916	0.648	0.655	0.511	0.929	0.761	0.711	0.651	0.966	0.782	0.790	0.674
	Tape	0.944	0.714	0.740	0.612	0.940	0.819	0.835	0.738	0.943	0.829	0.846	0.752
	Telephone	0.932	0.781	0.503	0.706	0.938	0.862	0.633	0.806	0.955	0.869	0.752	0.823
	Cola-0	0.771	0.565	0.521	0.432	0.884	0.678	0.603	0.573	0.900	0.787	0.768	0.700
	Cola-1	0.940	0.889	0.856	0.816	0.894	0.845	0.813	0.785	0.970	0.921	0.895	0.865
	Bowl	0.941	0.906	0.895	0.851	0.978	0.935	0.886	0.832	0.967	0.928	0.865	0.816
	Dice	0.840	0.815	0.786	0.721	0.925	0.891	0.763	0.726	0.971	0.899	0.843	0.816
	Teddy beer	0.896	0.798	0.768	0.665	0.893	0.792	0.773	0.712	0.951	0.884	0.843	0.752
<i>Chess</i>	TV-0	0.954	0.943	0.914	0.890	0.971	0.922	0.913	0.904	0.976	0.934	0.924	0.915
	TV-1	0.953	0.939	0.919	0.895	0.937	0.925	0.914	0.899	0.981	0.938	0.927	0.918
	Xbox-0	0.867	0.815	0.772	0.727	0.930	0.898	0.881	0.858	0.957	0.893	0.878	0.859
	Xbox-1	0.839	0.835	0.781	0.732	0.939	0.878	0.863	0.840	0.960	0.887	0.871	0.851
	Chair-0	0.902	0.879	0.840	0.802	0.955	0.911	0.899	0.882	0.976	0.922	0.908	0.890
	Chair-1	0.845	0.810	0.793	0.774	0.879	0.863	0.816	0.786	0.958	0.846	0.838	0.827
	Chair-2	0.884	0.882	0.858	0.832	0.948	0.873	0.866	0.855	0.895	0.863	0.853	0.842
	Timer	0.933	0.839	0.789	0.742	0.958	0.878	0.851	0.817	0.935	0.867	0.837	0.802
	Games	0.913	0.823	0.771	0.723	0.927	0.879	0.846	0.809	0.935	0.868	0.838	0.802
	Interruptor	0.934	0.695	0.600	0.531	0.941	0.778	0.695	0.626	0.947	0.785	0.697	0.625
	Gamepad	0.913	0.756	0.677	0.610	0.924	0.834	0.743	0.684	0.954	0.804	0.771	0.708

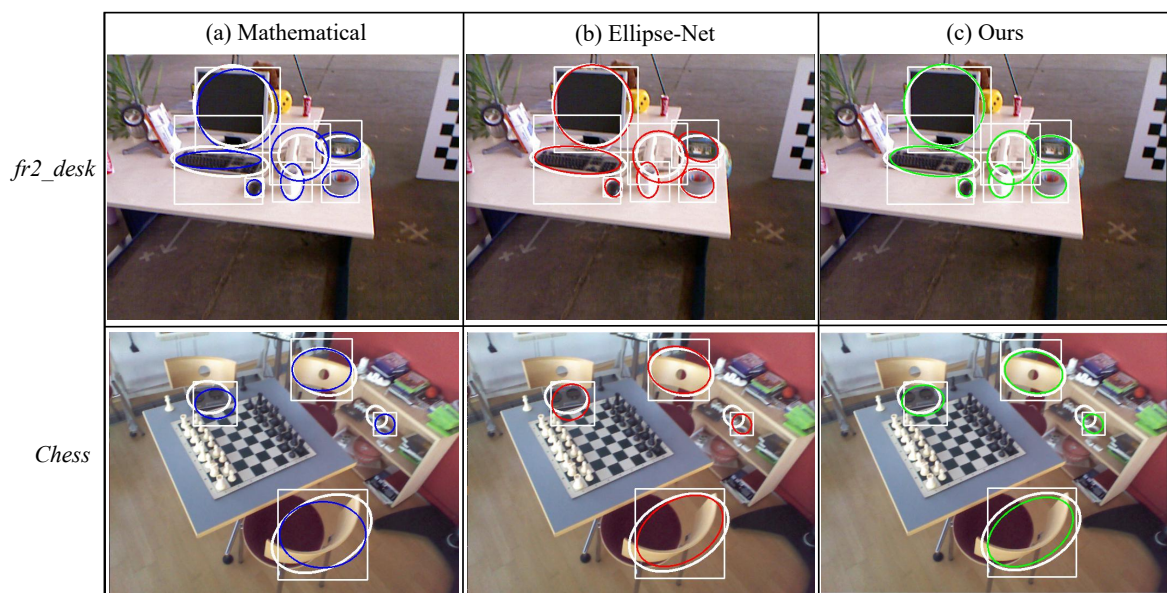


Figure 5. Different methods for ellipse prediction in bounding box with 15 pixel noise. The top is the predicted result of an image in the *fr2_desk*, and the bottom is the predicted result of an image in the *Chess*. (a–c) are comparisons between the three methods (Mathematical, Ellipse-Net and DEllipse-Net (Ours)) and ground truth, respectively. The white rectangle is the bounding box after adding 15 pixels of noise. “White ellipse” is the ground truth, “Blue ellipse” is Mathematical, “Red ellipse” is Ellipse-Net, and “Green ellipse” is DEllipse-Net.

Table 3. $ATE(m)$ and $ARE(^{\circ})$ of three ellipse prediction models under different bounding box noises. “Red” and “Blue” represent the best values of ATE and ARE , respectively.

Model	Errors	<i>fr2_desk</i>				<i>Chess</i>			
		0 pix	5 pix	10 pix	15 pix	0 pix	5 pix	10 pix	15 pix
Mathematical [15]	$ATE(m)$	0.037	0.053	0.100	0.190	0.078	0.087	0.162	0.303
	$ARE(^{\circ})$	1.36	1.89	3.47	7.10	2.54	2.73	5.14	10.32
Ellipse-Net [31]	$ATE(m)$	0.015	0.028	0.039	0.163	0.075	0.068	0.102	0.149
	$ARE(^{\circ})$	0.77	1.14	1.52	4.25	2.58	2.15	3.22	4.71
Ours	$ATE(m)$	0.012	0.021	0.046	0.077	0.055	0.066	0.095	0.144
	$ARE(^{\circ})$	0.56	0.90	1.72	2.70	1.86	2.03	2.89	4.48

More specifically, we present the statistical results of the IoU_{error} for different objects under undisturbed conditions, as shown in Figures 6 and 7, respectively. These line graphs illustrate the percentage of ellipses at various thresholds of IoU_{error} . In the case of the *fr2_desk* dataset (Figure 6), eight objects achieved optimal results at different thresholds. For instance, the “TV” had all predicted IoU_{error} below 0.08. When predicting the “Cup”, the threshold for IoU_{error} was set at 0.1, where our method achieved a 100% success rate, while Ellipse-Net accounted for 82% and Mathematical accounted for 80%. Furthermore, there are notable differences in ellipse prediction accuracy for different objects within the same category. For example, in the prediction of “Cola”, the Mathematical and Ellipse-Net exhibit varying predictive capabilities, while our approach demonstrates relatively high robustness. Similarly, as shown in Figure 7, our method also achieved the best results for *Chess*.

Finally, we compared the model sizes of the ellipses across the three methods and the average prediction time per frame, as shown in Table 4. Since the Mathematical relies on mathematical techniques for ellipse prediction, it does not have a traditional model,

resulting in minimal time required for predicting ellipses from image frames. In contrast, the single model size of Ellipse-Net reaches 82.6 MB, and each object requires corresponding model training during actual operation. Consequently, when dealing with a large number of objects in the environment, the space occupied becomes substantial, and it also takes the most time to predict ellipses for a single frame. However, DELLipse-Net only requires the training of one model to handle ellipse predictions for all objects, and our ellipse model is only 18.3 MB in size, with a running time that is shorter than that of Ellipse-Net. This demonstrates that DELLipse-Net is both lightweight and portable.

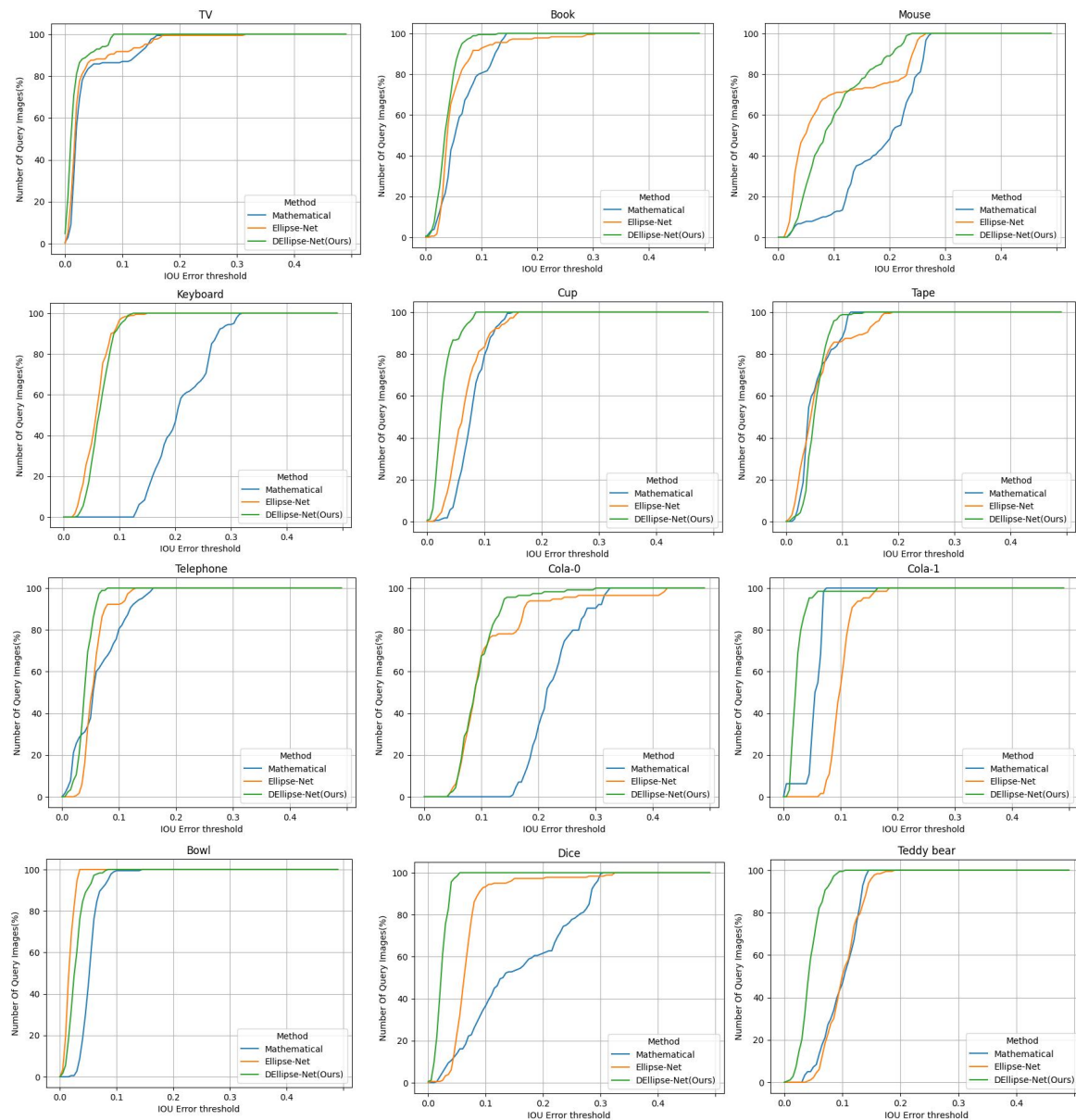


Figure 6. The percentage of images in *fr2_desk* with different IoU_{error} thresholds as a proportion of the total number of images.

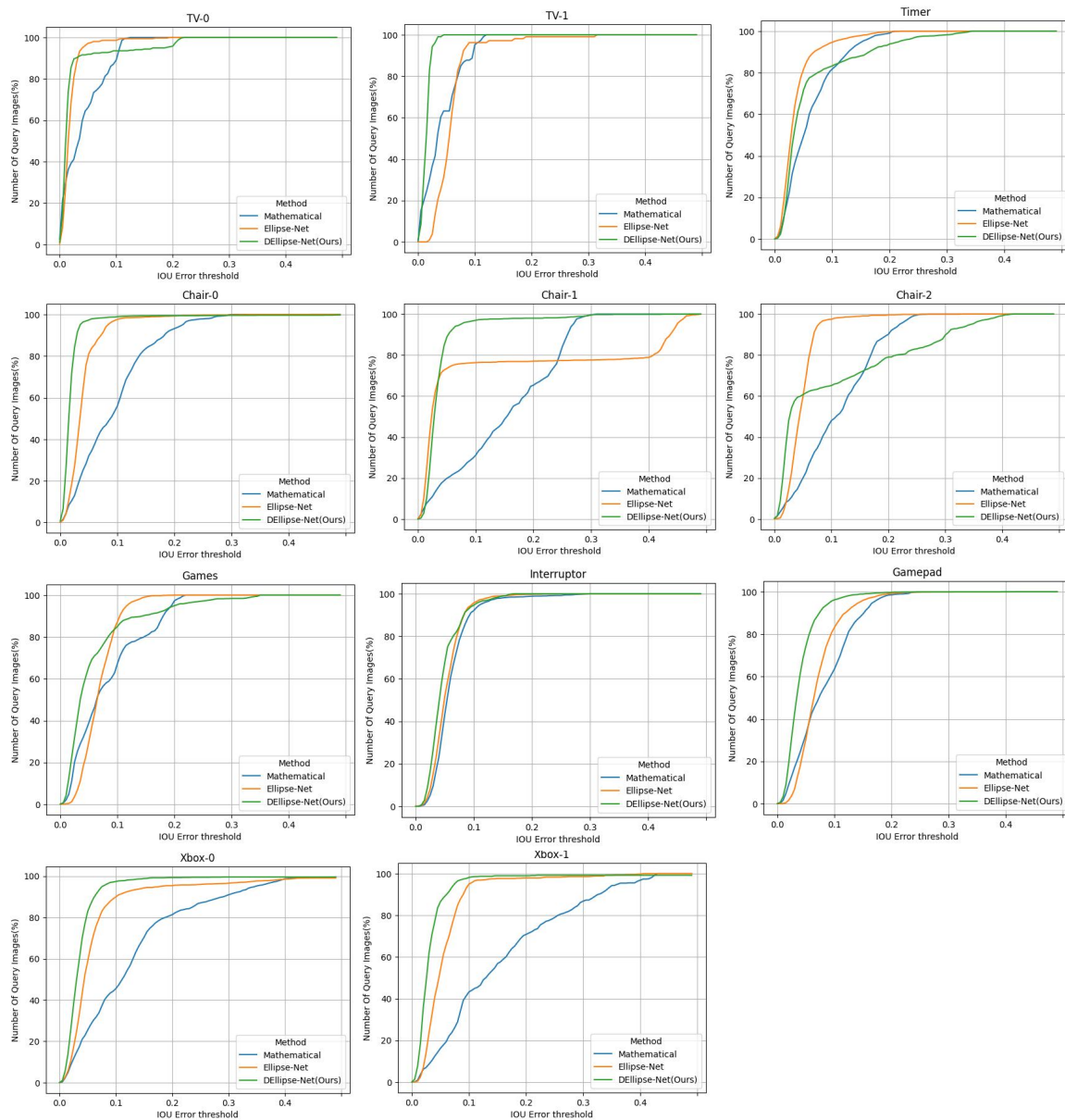


Figure 7. The percentage of images in *Chess* with different IoU_{error} thresholds as a proportion of the total number of images.

Table 4. The size and runtime of different models in the *Chess* sequence. Runtime is the average running time of an image.

Model	Model Size (MB)	Model Number	Total Size (MB)	Runtime (s)
Mathematical [15]	-	-	-	0.00014
Ellipse-Net [31]	82.6	11	908.6	0.01688
DEllipse-Net	18.3	1	18.3	0.01087

5.2. Localization Accuracy with Object-Level Instance Topology-Alignment

Localization prediction involves matching the predicted ellipse in the image with the corresponding ellipsoid in the environment. In previous works, such as [15,31], ellipses and ellipsoids are matched by category, where a single ellipse may correspond to the projection of multiple ellipsoids, leading to pose estimation errors. By applying our object-level

instance topology to ellipse-ellipsoid localization, we achieve more accurate matching between ellipses and ellipsoids.

We use the *Chess* dataset from 7-Scenes [34] for our experiments. The *Chess* dataset contains 6 sequences, each with 1000 images. Since our method requires the detection of two or more ellipse-ellipsoid pairs for pose estimation, we use this criterion for image filtering. The primary aim of this experiment is to evaluate the impact of our instance topology on pose estimation. Therefore, we conducted localization tests using three different ellipse estimation models under two different ellipse-ellipsoid matching modes. *ATE* and *ARE* are used as metrics to assess localization accuracy.

As shown in Table 5, “×” indicates that instance topology is not used, while “√” indicates the use of our instance topology. In the “×” mode, Ours effectively reduces both translation and rotation errors. In the translation test, our method achieved the lowest *ATE* in four sequences, while it ranked second in seq-04 and seq-06. In the rotation test, our method produced suboptimal results in seq-01 and seq-02, but achieved the lowest *ARE* in the remaining sequences. In the “√” mode, our method demonstrated similar performance.

Table 5. The impact of different ellipse prediction methods and alignment methods on localization. “√” used object-level instance topology and alignment methods, while “×” did not use them. Translation error is *ATE* (m), Rotation error is *ARE* (°).

Ellipse Model	Errors	Topology	seq-01	seq-02	seq-03	seq-04	seq-05	seq-06
Mathematical [15]	<i>ATE</i> (m)	×	0.147	0.152	0.223	0.208	0.162	0.082
		√	0.066	0.053	0.037	0.036	0.035	0.045
	<i>ARE</i> (°)	×	5.45	6.59	10.51	8.88	6.48	2.96
		√	2.18	2.04	1.07	1.03	1.19	1.55
Ellipse-Net [31]	<i>ATE</i> (m)	×	0.136	0.124	0.165	0.164	0.108	0.048
		√	0.080	0.048	0.052	0.037	0.041	0.046
	<i>ARE</i> (°)	×	6.44	4.68	7.79	7.62	4.42	1.60
		√	2.80	2.35	1.71	1.09	1.46	1.60
DEllipse-Net(Ours)	<i>ATE</i> (m)	×	0.125	0.123	0.162	0.165	0.105	0.049
		√	0.050	0.042	0.036	0.032	0.049	0.042
	<i>ARE</i> (°)	×	5.77	5.12	7.50	7.31	4.32	1.65
		√	1.67	1.92	1.08	0.95	1.75	1.47

Compared with the “×” mode, after using our instance topology, the translation and rotation errors of the camera are greatly reduced. In seq-04 sequence, Mathematical reduces the translation error from 0.208 m to 0.036 m and the corresponding rotation error from 8.88° to 1.03° after using descriptor matching. The translation error of Ellipse-Net decreased from 0.164 m to 0.037 m, and the rotation error decreased from 7.62° to 1.09°. The translation error of our method decreased from 0.165 m to 0.032 m, and the rotation error decreased from 7.31° to 0.95°. This shows that our instance topology greatly improves the probability of object matching, thereby improving the localization accuracy.

5.3. Pose Accuracy in Different Lighting Environments

When performing localization in indoor environments, the accuracy is affected by various factors due to the time gap between map creation and subsequent localization. Among these factors, lighting variation has the most significant impact on visual localization. In this section, we compare a feature-based method, OVS (OpenVSLAM) [35], with two Ellipse-Ellipsoid-based methods (Ellipse-Net and ours) to evaluate the localization performance of these three approaches under different lighting conditions. We use the *fr2_desk*, *fr3_office_household*, *Chess*, and *Redkitchen* sequences as test data and adjust the HSV values in the images to simulate low-light conditions, as shown in Figure 8. Under normal lighting conditions, we use the RGB-D model of the full OVS system for map

creation, but only its relocalization module during the localization phase. Ellipse-Net follows the approach provided by [31], while our method is based on DEllipse-Net and object-level instances topology-alignment localization. *ATE* and *ARE* are used as metrics to evaluate localization accuracy.

As shown in Table 6, the localization accuracy of all three methods decreases in dark environments compared to normal lighting. In the *fr2_desk* sequence, under normal lighting, OVS has translation and rotation errors of 0.102 m and 8.73°, Ellipse-Net has errors of 0.024 m and 0.89°, and our method achieves 0.012 m and 0.56°. In dark conditions, OVS's errors rise to 0.119 m and 6.39°, Ellipse-Net's to 0.065 m and 2.44°, while our method shows only 0.022 m and 0.89°. Overall, our method consistently outperforms OVS and Ellipse-Net across all lighting conditions, with similar advantages in the other three environments.

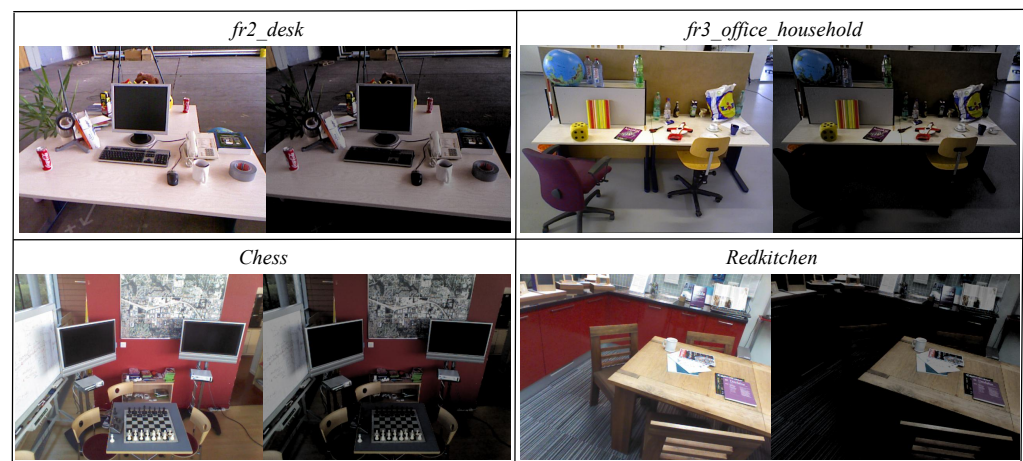


Figure 8. A comparison of images from the *fr2_desk*, *fr3_office_household*, *Chess*, and *Redkitchen* sequences under normal and dark conditions.

Table 6. The *ATE* and *ARE* of OVS, Ellipse-Net, and our method across four sequences under normal and dark conditions.

Method	Errors	<i>fr2_desk</i>		<i>fr3_office_household</i>		<i>Chess</i>		<i>Redkitchen</i>	
		Normal	Dark	Normal	Dark	Normal	Dark	Normal	Dark
OVS [35]	<i>ATE</i> (m)	0.102	0.119	0.044	0.067	0.081	0.129	0.040	0.050
	<i>ARE</i> (°)	8.73	6.39	1.15	1.70	9.59	10.65	1.59	2.10
Ellipse-Net [31]	<i>ATE</i> (m)	0.024	0.065	0.049	0.088	0.143	0.206	0.043	0.107
	<i>ARE</i> (°)	0.89	2.44	1.57	2.41	6.88	10.13	1.32	3.23
Ours	<i>ATE</i> (m)	0.012	0.022	0.013	0.021	0.055	0.110	0.037	0.043
	<i>ARE</i> (°)	0.56	0.89	0.33	0.52	1.86	3.93	1.61	2.02

Despite both Ellipse-Net and our approach being based on the Ellipse-Ellipsoid model, our method is more robust to lighting variations. Across all four sequences, Ellipse-Net shows significant localization accuracy changes under different lighting conditions. For example, in the *fr3_office_household* sequence, its translation and rotation errors increase by 0.039 m and 0.84° in dark conditions, while OVS's increase by 0.023 m and 0.45°. In contrast, our method sees only minimal increases of 0.009 m and 0.19°, demonstrating its robustness against lighting variations.

5.4. Pose Accuracy When Objects Are Occluded

This experiment tests the localization ability of classical point-based methods and ellipsoid-ellipsoid methods under object occlusion. In four sequences, to simulate occlusion,

we applied a black mask to occlude corresponding objects in the RGB images. We sequentially occluded 1, 2, and 3 objects in the query images to evaluate how both methods handle object occlusion, as shown in Figure 9. To ensure the ellipse-ellipsoid model can still predict poses, we selected query images containing five or more objects from the sequences. We used images without occlusions for map construction and then performed localization testing using the occluded images. The localization performance of both methods under occlusion was evaluated using *ATE* and *ARE*.

As shown in Table 7, in the *fr2_desk* sequence, when the number of occluded objects increased from one to two, OVS's translation error rose slightly from 0.238 m to 0.241 m, and its rotation error from 17.0° to 17.08° . When the number of occlusions increased from two to three, the translation error increased to 0.260 m, and the rotation error to 18.19° . Ellipse-Net showed a similar trend, with translation errors increasing by 0.002 m and 0.016 m, and rotation errors by 0.01° and 0.29° . In contrast, our method showed no increase in errors during the first stage and only a small rise of 0.017 m in translation and 0.49° in rotation errors during the second stage. These results suggest that localization accuracy decreases with more occlusions for OVS, Ellipse-Net, and our method. This trend was also observed in the *fr3_office_household*, *Redkitchen*, and *Chess* sequences.

The Ellipse-Ellipsoid model relies on object detection for ellipse prediction, matching, and global localization. When landmarks are occluded, both Ellipse-Net and our method fail to detect them, leading to localization failure. In contrast, OVS uses ORB feature points, matched with keyframes via BoW [16] and applies PnP for pose estimation, making it less sensitive to occlusion after successful matching. During testing, we used black ellipses to occlude objects, preventing feature detection in darkened areas and reducing incorrect matches. Although our method is more affected by occlusions, it still outperforms OVS, demonstrating superior localization accuracy under occlusion conditions.

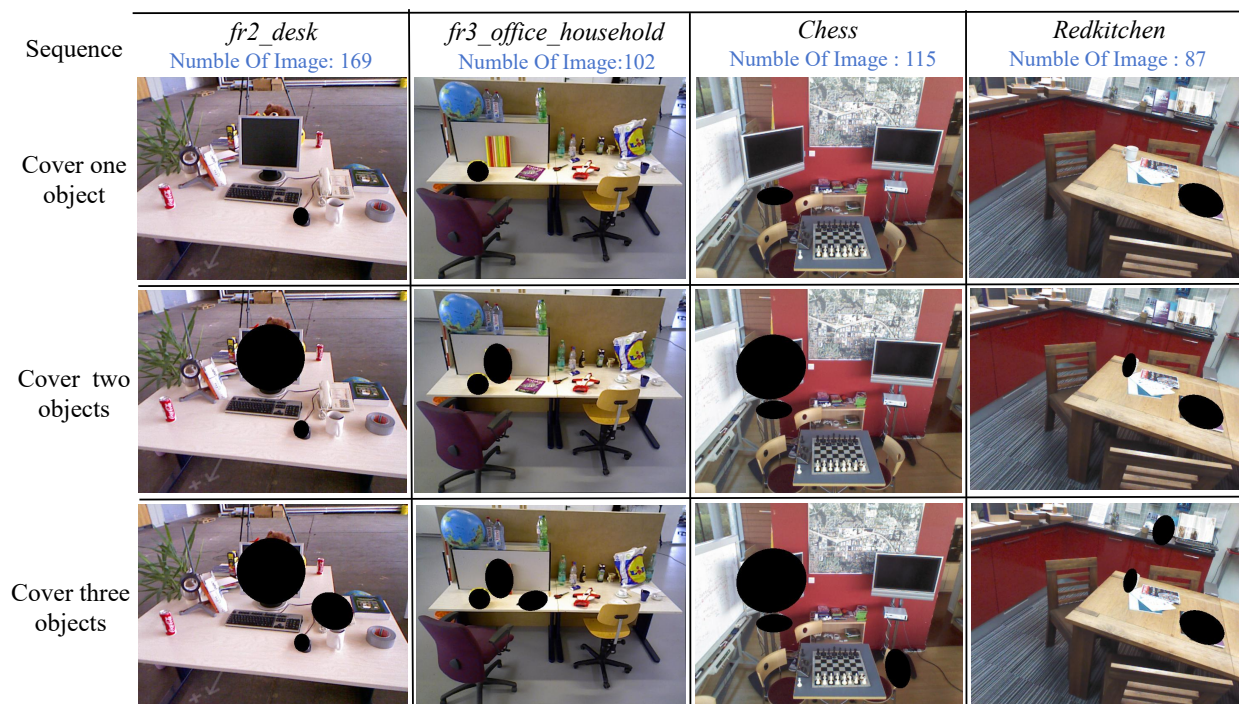


Figure 9. The image performance of the *fr2_desk*, *fr3_office_household*, *Chess*, and *Redkitchen* sequences with one, two, and three occlusions.

Table 7. ATE and ARE of OVS, Ellipse-Net, and Ours across four sequences with varying numbers of occlusions.

Method	Cover Number	<i>fr2_desk</i>		<i>fr3_office_household</i>		<i>Chess</i>		<i>Redkitchen</i>	
		ATE (m)	ARE (°)	ATE (m)	ARE (°)	ATE (m)	ARE (°)	ATE (m)	ARE (°)
OVS [35]	One	0.238	17.00	0.046	1.04	0.053	1.67	0.022	0.96
	Two	0.241	17.08	0.045	0.99	0.062	2.00	0.026	1.08
	Three	0.260	18.19	0.051	1.39	0.069	2.23	0.029	1.15
Ellipse-Net [31]	One	0.030	1.32	0.029	0.86	0.095	2.94	0.023	0.98
	Two	0.032	1.33	0.037	1.01	0.098	3.04	0.026	1.04
	Three	0.048	1.62	0.044	1.49	0.112	3.33	0.042	1.31
Ours	One	0.017	0.83	0.021	0.46	0.031	0.80	0.011	0.32
	Two	0.017	0.83	0.022	0.49	0.038	1.31	0.013	0.43
	Three	0.034	1.32	0.037	0.88	0.042	1.42	0.026	1.28

6. Conclusions

We propose an object-based global localization method that estimates the 6-DoF camera pose using an Ellipse-Ellipsoid model and object-level instance topology alignment. First, we designed a neural network that predicts ellipses using RGB and depth images. Second, the instance topology and alignment model allows for successful matching between detected ellipses and those in the environment. Finally, we achieve accurate and robust localization in a 3D ellipsoid environment that we constructed. We evaluated our method on publicly available and pre-processed datasets. Compared to existing approaches, our method can robustly predict ellipses in images and, combined with object-level topology alignment, maintains high localization accuracy even in environments with significant changes in lighting and occlusion.

However, our method has some limitations. For instance, we have not conducted experiments in real-world environments for lighting and occlusion, and the ellipsoid map still requires manual annotation. In future work, when experimental facilities are fully equipped, we plan to conduct tests in real-world settings and further optimize our algorithm. Additionally, we are working on the construction of instance and ellipsoid maps, aiming to simultaneously build both instance and ellipsoid maps.

Author Contributions: Methodology, H.W.; Project administration, Y.L.; Software, H.W.; Supervision, Y.L.; Writing & original draft, H.W.; Writing—review & editing, C.W. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Key Special Projects of Heilongjiang Province’s Key R&D Program (No. 2023ZX01A01) and Heilongjiang Province’s Key R&D Program: ‘Leading the Charge with Open Competition’ (No. 2023ZXJ01A02).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, K.; Zhang, J.; Liu, J.; Tong, Q.; Liu, R.; Chen, S. Semantic Visual Simultaneous Localization and Mapping: A Survey. *arXiv* **2022**, arXiv:2209.06428. [[CrossRef](#)]
- Yin, Z.; Wen, H.; Nie, W.; Zhou, M. Localization of Mobile Robots Based on Depth Camera. *Remote Sens.* **2023**, *15*, 4016. [[CrossRef](#)]
- Huang, Y.; Xie, F.; Zhao, J.; Gao, Z.; Chen, J.; Zhao, F.; Liu, X. ULG-SLAM: A Novel Unsupervised Learning and Geometric Feature-Based Visual SLAM Algorithm for Robot Localizability Estimation. *Remote Sens.* **2024**, *16*, 1968. [[CrossRef](#)]
- Wang, Y.; Jiang, C.; Chen, X. GOREloc: Graph-Based Object-Level Relocalization for Visual SLAM. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8234–8241. [[CrossRef](#)]

5. Zhao, X.; Li, Q.; Wang, C.; Dou, H.; Liu, B. Robust Depth-Aided RGBD-Inertial Odometry for Indoor Localization. *Measurement* **2023**, *209*, 112487. [\[CrossRef\]](#)
6. Rosinol, A.; Abate, M.; Chang, Y.; Carlone, L.. Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 1689–1696.
7. Abate, M.; Chang, Y.; Hughes, N.; Carlone, L. Kimera2: Robust and Accurate Metric-Semantic SLAM in the Real World. In *Springer Proceedings in Advanced Robotics*; Springer: Cham, Switzerland, 2024. [8. \[CrossRef\]](#)
8. Li, M.; Ma, Y.; Qiu, Q. SemanticSLAM: Learning based Semantic Map Construction and Robust Camera Localization. In Proceedings of the 2023 IEEE Symposium Series on Computational Intelligence (SSCI), Mexico City, Mexico, 5–8 December 2023; pp. 312–317. [\[CrossRef\]](#)
9. Adkins, A.; Chen, T.; Biswas, J. ObVi-SLAM: Long-Term Object-Visual SLAM. *IEEE Robot. Autom. Lett.* **2024**, *9*, 2909–2916. [\[CrossRef\]](#)
10. Guo, X.; Hu, J.; Chen, J.; Deng, F.; Lam, T.L. Semantic Histogram Based Graph Matching for Real-Time Multi-Robot Global Localization in Large Scale Environment. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1–8. [\[CrossRef\]](#)
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [\[CrossRef\]](#)
13. Jocher, G. Ultralytics YOLO. *arXiv e-prints* **2023**. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 22 February 2024).
14. Li, J.; Meger, D.; Dudek, G. Context-coherent scenes of objects for camera pose estimation. In Proceedings of the 2017 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.
15. Gaudillière, V.; Simon, G.; Berger, M.-O. Perspective-2-Ellipsoid: Bridging the Gap Between Object Detections and 6-DoF Camera Pose. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5189–5196. [\[CrossRef\]](#)
16. McCallum, A.K. Bow: A toolkit for statistical language modeling. In Proceedings of the ICASSP, Atlanta, GA, USA, 7–10 May 1996.
17. Cummins, M.; Newman, P. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665. [\[CrossRef\]](#)
18. Chow, C.; Liu, C. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory* **1968**, *14*, 462–467. [\[CrossRef\]](#)
19. Galvez-Lopez, D.; Tardos, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Robot.* **2012**, *28*, 1188–1197. [\[CrossRef\]](#)
20. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [\[CrossRef\]](#)
21. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [\[CrossRef\]](#)
22. Campos, C.; Elvira, R.; Gomez Rodriguez, J.J.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [\[CrossRef\]](#)
23. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [\[CrossRef\]](#)
24. Sarlin, P.-E.; Cadena, C.; Siegwart, R.; Dymczyk, M. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019; pp. 12708–12717. [\[CrossRef\]](#)
25. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-Free Local Feature Matching with Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8918–8927. [\[CrossRef\]](#)
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
27. Huang, Z.; Zhou, H.; Li, Y.; Yang, B.; Xu, Y.; Zhou, X.; Bao, H.; Zhang, G.; Li, H. VS-Net: Voting with Segmentation for Visual Localization. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 6097–6107. [\[CrossRef\]](#)
28. Liu, Y.; Petillot, Y.; Lane, D.; Wang, S. Global Localization with Object-Level Semantics and Topology. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4909–4915. [\[CrossRef\]](#)
29. Lin, S.; Wang, J.; Xu, M.; Zhao, H.; Chen, Z. Topology Aware Object-Level Semantic Mapping Towards More Robust Loop Closure. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7041–7048. [\[CrossRef\]](#)
30. Gaudillière, V.; Simon, G.; Berger, M.-O. Camera Relocalization with Ellipsoidal Abstraction of Objects. In Proceedings of the 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Beijing, China, 14–18 October 2019; pp. 8–18. [\[CrossRef\]](#)

31. Zins, M.; Simon, G.; Berger, M.O. Object-Based Visual Camera Pose Estimation From Ellipsoidal Model and 3D-Aware Ellipse Prediction. *Int. J. Comput. Vis.* **2022**, *130*, 1107–1126. [[CrossRef](#)]
32. Oleynikova, H.; Taylor, Z.; Fehr, M.; Siegwart, R.; Nieto, J. Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1366–1373. [[CrossRef](#)]
33. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura-Algarve, Portugal, 7–12 October 2012.
34. Glocker, B.; Izadi, S.; Shotton, J.; Criminisi, A. Real-time RGB-D camera relocalization. In Proceedings of the 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, SA, Australia, 1–4 October 2013; pp. 173–179. [[CrossRef](#)]
35. Sumikura, S.; Shibuya, M.; Sakurada, K. OpenVSLAM: A Versatile Visual SLAM Framework. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.