2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

HSS-SLAM: Human-in-the-Loop Semantic SLAM Represented by Superquadrics

Yulong Li[1], Yunzhou Zhang[1]*, Bin Zhao[1], Zhiyao Zhang[1], You Shen[1], Tengda Zhang[1], Guolu Chen[1]

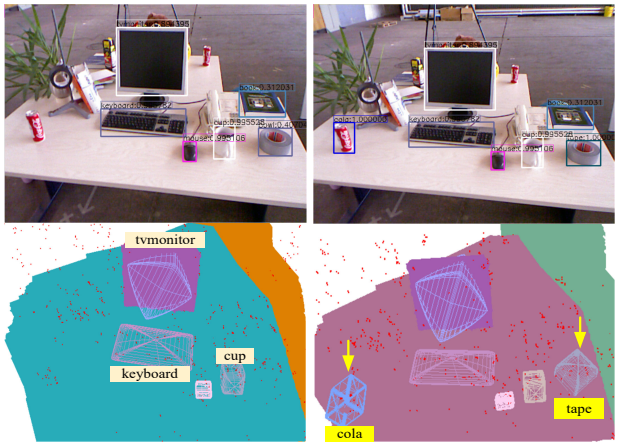
***Abstract*— The advancement of object detection algorithms has catalyzed the development of object-level semantic SLAM. However, due to missed and false detections, object-level semantic SLAM fails to represent the objects within the scene adequately. Therefore, this paper proposes a novel object-level semantic SLAM termed HSS-SLAM. We incorporate human-in-the-loop into our method, establishing an interaction module to facilitate human editing and rectifying semantic information. Additionally, to minimize the manual correction workload, a lightweight and intuitive method for semantic extension is proposed, augmenting the semantic richness of the global map with a few operations. Furthermore, our method adopts superquadrics for object representation, enabling detailed descriptions of various object shapes. This mitigates the limitation of conventional semantic mapping, where objects are difficult to distinguish due to the reliance on a single-shape representation. Subsequently, precise estimation of superquadric parameters and camera poses is achieved through joint optimization. Extensive experiments conducted on TUM RGB-D and Scenes V2 datasets demonstrate that the proposed approach exhibits competitive performance, surpassing current methods in both object representation and camera localization accuracy.**

I. Introduction

Conventional SLAM methodologies leverage geometric features, such as points, lines, and planes, for constructing point cloud maps [1], [2] or plane maps [3] representing the scene. While these methods enhance environmental representation to some extent, they fall short in adequately addressing the needs of indoor robots necessitating interaction with objects within their environment.

In contrast to the feature-point method, object-level SLAM employs geometric models such as CAD models [4], cubes [5], and quadrics [6] for object representation. Object-level semantic SLAM not only aids in robot localization but also enhances map reusability and object recognition within the environment, thus augmenting the robot's capability for environmental interaction. Among various forms of object representation, quadrics are extensively employed in SLAM [7]–[9] owing to their compact expression form and direct projection model. However, the simplistic shape of quadrics limits their ability to express various objects within the environment in detail accurately.

Currently, object detection algorithms have emerged as the predominant approach for acquiring semantics in traditional object-level SLAM. Conventional object-level SLAM necessitates multi-frame observations from diverse angles to initialize object representation. However, owing to missed and false detections across various scenes, the semantic information obtained through the object detection algorithm is not accurate enough, resulting in the loss of semantic information in the map and the omission of object representation. Consequently, the robot's ability to perceive and interact with the environment is significantly compromised. However, from a human perspective, these errors can be easily corrected through guidance or interaction. In fact, some efforts have applied human-in-the-loop methodologies to SLAM, yielding preliminary results in closed-loop detection [10] and human-machine interaction [11], but the utilization of human-in-the-loop for rectifying missed and erroneous semantic information has not yet garnered adequate attention.



(a) original map (b) corrected map

Fig. 1. **Semantic correction in the map**. On the left is the map constructed using original semantic information. On the right, a portion of missed (cola) and erroneous (tape) semantic information has been corrected, enabling the representation of previously missed objects in the map.

In this paper, we propose HSS-SLAM, a human-in-the-loop object-level semantic SLAM employing superquadric representations, which effectively mitigates the deficiency of semantic information in maps and the challenge of differentiating objects using quadrics, as illustrated in Fig.1. Our approach combines planes and superquadrics to effectively express spatial structure characteristics of the environment while representing objects in space. We integrate human-in-the-loop functionality into our method to facilitate online interaction. Users can interactively edit and correct missed detections and false labels in the image detection results. Subsequently, superquadric initialization is conducted through detection box projection and fitting to map points. Simultane-

*The corresponding author of this paper
[1]Yulong Li, Yunzhou Zhang, Bin Zhao, Zhiyao Zhang, You Shen, Tengda Zhang, and Guolu Chen are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China. zhangyunzhou@mail.neu.edu.cn

This work was supported by National Natural Science Foundation of China (No. 61973066) and Major Science and Technology Projects of Liaoning Province(No. 2021JH1/10400049).

ously, we improve the pose of superquadrics and the overall system localization accuracy through joint optimization. We conduct experiments on public datasets, achieving state-of-the-art performance in terms of object representation and camera localization accuracy.

The main contributions of this work are as follows:

- We construct a human-in-the-loop interaction module to facilitate interactive editing and correction of semantic information, thereby providing the system with accurate semantic information input. To the best of our knowledge, we are the first to employ human-in-the-loop for correcting semantic information.
- We introduce a lightweight and straightforward strategy for expanding correction results, which extends the spatially continuous range of the manual refinement effects achieved through human-in-the-loop interaction.
- We estimate the pose parameters and shape parameters of superquadric by using the semantic information in the image and the minimum convex hull constructed by the map points respectively. This process achieves a rapid initialization of objects represented by superquadrics.
- We design HSS-SLAM, a comprehensive visual semantic SLAM system. Through extensive experiments on public datasets, we validate that HSS-SLAM enriches semantic information in maps, and achieves state-of-the-art performance in terms of object representation and the precision of localization and mapping.

II. Related Works

A. Object-level SLAM

Representing objects with appropriate models not only delineates object distribution in the map, improves the readability of the map, but also strengthens the interaction between the robot and the environment. In previous explorations of object representation, CAD models [4] were initially used to represent objects but proved unsuitable for unknown environments. CubeSLAM [5], as the first instance of object-level representation, transforms semantic information into three-dimensional cubes to represent objects. Subsequently, the method of representing objects with quadrics [6] is found to have a compact mathematical expression, which is more suitable for model updates in practice. This not only makes quadrics the mainstream approach to express objects [12]–[14] but also promotes the improvement of object-level semantic SLAM system framework [15], [16]. Apart from accurately representing objects in the environment, some works [17]–[19] improve accuracy by incorporating planes in the structural scene, and OA-SLAM [20] further applies the expressed objects in relocalization, maximizing the utilization of object-level maps. Recently, to facilitate grasping in diverse objects [21] and recovering the original shapes of various 3D data types within the scene [22], superquadric has gradually been applied in object representation [23]. We represent object detection results using superquadrics to enhance object distinguishability in the map.

B. Human-in-the-Loop SLAM

Human-in-the-loop is employed to delineate a novel form of interaction between humans and machine learning [24]. In SLAM, when confronted with large-scale scenes, numerous small errors may manifest in the map, and optimizing them solely through algorithms can result in increased computational complexity and time consumption. HitL-SLAM [10] integrates sparse human correction, accelerating the construction of large-scale maps by optimizing specified relationships among map segments. A-SLAM [11] allows real-time user interaction, guiding the robot to correct pose and map errors during SLAM operation, but it is limited to two-dimensional plane maps. Subsequently, Ouyang *et al*. [25] introduced human-in-the-loop to point cloud segmentation models, training them with artificial semantics to enhance localization accuracy. In our approach, we leverage human-in-the-loop to manually refine semantic information, thereby constructing object-level maps with rich semantic details.

III. System overview

The HSS-SLAM presented in this paper is modified based on ORB-SLAM2 [1], and the system architecture is illustrated in Fig.2. In the tracking thread, integration occurs among a human-in-the-loop interaction module, an object detection module, and a plane extraction module. RGB and depth images received as input are respectively fed into the object detection and plane extraction modules. By monitoring the results of object detection in the current frame, the human-in-the-loop interaction module can be manually activated to select and edit areas exhibiting semantic information errors or omissions, thereby facilitating corrective actions. The pose is estimated through plane constraints and feature point constraints. Subsequently, a convex hull is constructed for the map points corresponding to the detection box, and the parameters controlling the shape of the superquadric are estimated to complete the initialization. Combined with the estimated camera pose, the object detection and plane extraction results of the current frame are associated with the constructed map. In the local mapping thread, joint optimization is conducted on camera poses, map points, plane landmarks, and superquadric landmarks. This comprehensive optimization process results in an environmental map encompassing geometric and semantic information.

IV. Method

A. Human-in-the-loop Semantic Refinement

Currently, in object-level semantic SLAM systems, semantic information and object representation primarily rely on the results of object detection algorithms. However, when object detection algorithms are applied to a new environment without additional training or parameter tuning, the probability of missed and false detection results significantly increases. This will increase the challenge for the system to build accurate semantic maps. The introduction of text or language guidance during detection can enhance the effectiveness of detection and reduce error rates. Additionally,

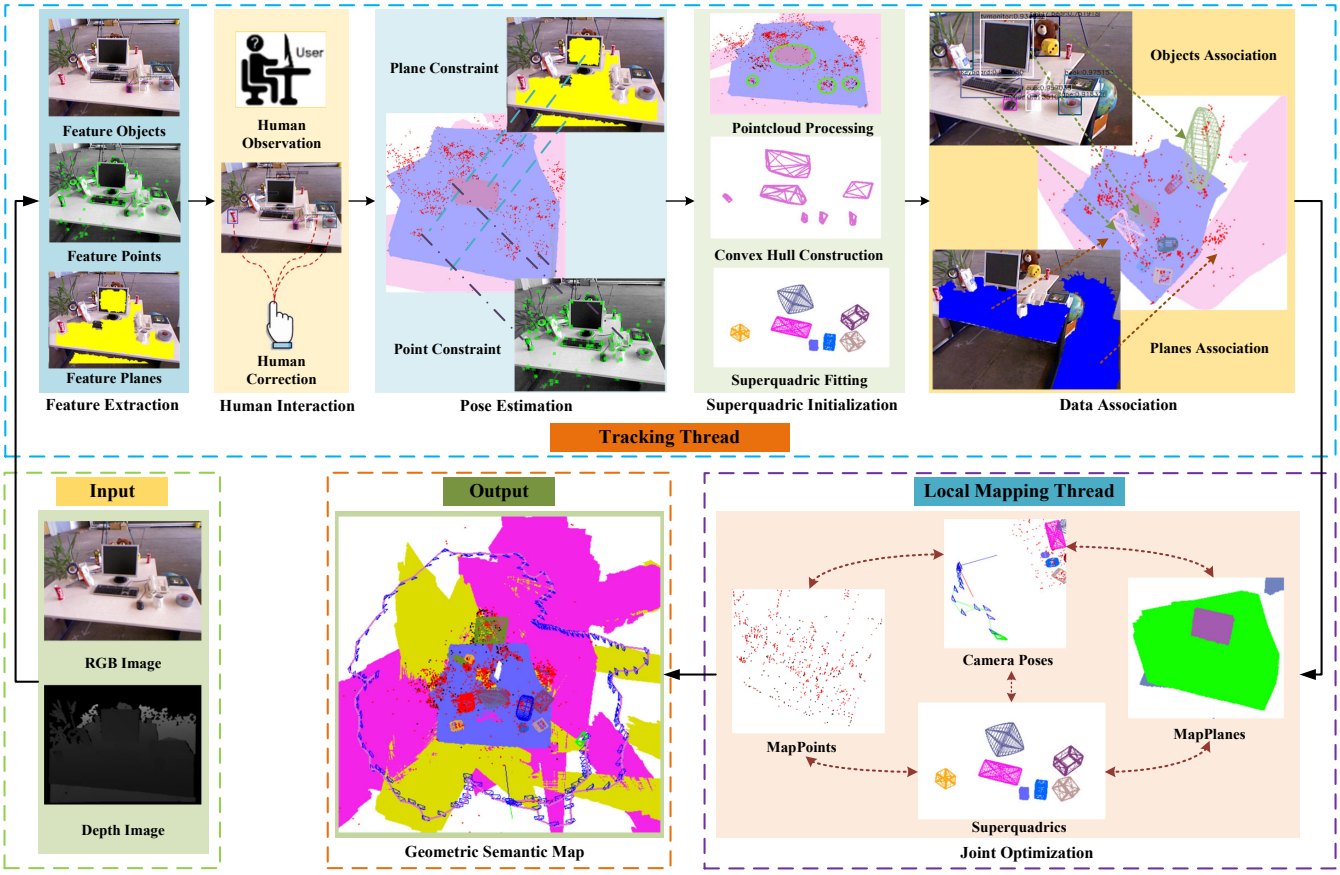


Fig. 2. **Overview of the proposed system.** The system modifies two threads. The tracking thread performs feature extraction, plane extraction, object detection, human-in-the-loop semantic refinement, pose estimation, superquadric initialization, and data association. The local mapping thread increases joint optimization to improve localization accuracy.

although SLAM systems can construct maps of the perceived environment, robots cannot autonomously complete this task entirely. Human involvement is still required to make judgments about the rationality of the mapping process.

Based on these two points, we explicitly incorporate the human role into the SLAM system. We have designed an online interactive module in the system, which can correct the semantic information obtained by semantic thread online. Upon input of RGB images into the system, YOLOv5 is used for detection, and users can visually observe the detection results. In cases of severe missed or false detections in the original results, users can suspend the system and enable the interactive correction module. By selecting objects in boxes and inputting labels to refine the image detection results, users then feed them back into the system to enhance the accuracy of object representation in the map concerning the environment. Fig.3 illustrates the complete process of correcting image detection results.

B. Local Homography

The camera model based on constant velocity motion exhibits a high degree of image overlap in a short period. To avoid repetitively correcting for similar scene and provide a sufficient number of frames for initialization, we have devised a simple extension method using local homography to expand the influence range of correction results. Homography can describe the pixel correspondence relationship on the same planar surface in space. The computation of the homography matrix in the SLAM system occurs after incorporating semantic information. This conflicts with our use of the homography matrix to expand results. Therefore, we calculate the local homography between two frames using the pixel points corresponding to the detection boxes. This local homography helps find corresponding detection boxes or pixel-level coordinates for manually corrected detection boxes in subsequent images with high redundancy.

We use H_{D_k} and O_{D_k} to represent the set of manual correction detection results and the set of original detection results obtained after correction in image frame I_k, respectively. The corresponding detection boxes in sets O_{D_k} and $O_{D_{k+1}}$ can be found using 2D-IoU(Intersection over Union). We consider the vertices of the matched detection boxes as corresponding point pairs, which are then used to calculate the local homography matrix $\mathbf{H_L}$. In mathematics, matrix $\mathbf{H_L}$ can be represented by a linear expression:

$$\mathbf{x}_i' = \mathbf{H_L}\mathbf{x}_i \tag{1}$$

where $\mathbf{x}_i = [x_i, y_i, w_i]^\top$ and $\mathbf{x}_i' = [x_i', y_i', w_i']^\top$ are a pair of matched points of $O^i_{D_k}$ and $O^i_{D_{k+1}}$ respectively. The equation is represented in vector form as the product:

$$\mathbf{x}_i' \times \mathbf{H_L}\mathbf{x}_i = \begin{bmatrix} y_i'\mathbf{h_L}^{3\mathrm{T}}\mathbf{x}_i - w_i'\mathbf{h_L}^{2\mathrm{T}}\mathbf{x}_i \\ w_i'\mathbf{h_L}^{1\mathrm{T}}\mathbf{x}_i - x_i'\mathbf{h_L}^{3\mathrm{T}}\mathbf{x}_i \\ x_i'\mathbf{h_L}^{2\mathrm{T}}\mathbf{x}_i - y_i'\mathbf{h_L}^{1\mathrm{T}}\mathbf{x}_i \end{bmatrix} = 0 \tag{2}$$

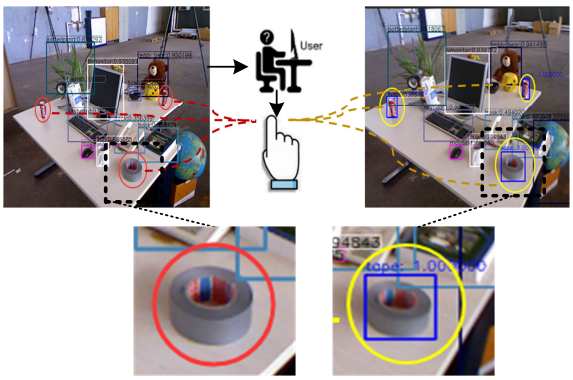


Fig. 3. The user performs online correction on the image semantics, where red represents omissions, and blue indicates the refined results. The refined results can be treated as ground truth, with the confidence set to 1.

where $\mathbf{h}_{\mathbf{L}}^{jT}$ is the j-th row of $\mathbf{H_L}$, which also can be represented in matrix form:

$$\begin{bmatrix} 0^\top & -w_i'\mathbf{x}_i^\top & y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & 0^\top & -x_i'\mathbf{x}_i^\top \\ -y_i'\mathbf{x}_i^\top & x_i'\mathbf{x}_i^\top & 0^\top \end{bmatrix} \begin{bmatrix} \mathbf{h}_{\mathbf{L}}^1 \\ \mathbf{h}_{\mathbf{L}}^2 \\ \mathbf{h}_{\mathbf{L}}^3 \end{bmatrix} = 0 \tag{3}$$

Then we can express it as a linear expression $\mathbf{Ah} = 0$, where $\mathbf{A}$ is a $2n \times 9$ matrix, with n as the number of point pairs. The nine elements of matrix $\mathbf{H_L}$ can be estimated through the following method:

$$\hat{\mathbf{h}}_{\mathbf{L}} = arg\min_{\mathbf{h_L}} ||\mathbf{Ah_L}||^2 \tag{4}$$

In practical calculations, there should be at least four point pairs, which implies that having just one pair of detection boxes is sufficient to meet the minimum necessary geometric constraints. Once $\mathbf{H_L}$ is obtained, the corrected coordinates of detection boxes in the next frame can be obtained through $H^i_{D_{k+1}} = \mathbf{H_L} H^i_{D_k}$. Subsequently, $H_{D_{k+1}}$ is merged with $O_{D_{k+1}}$ within frame I_{k+1}, achieving semantic automatic correction in frame I_{k+1}. In Alg.1, we provide a detailed and comprehensive demonstration of the entire process for extending the scope of semantic fine-tuning results.

C. Superquadric Initialization

Superquadric is an extension of a quadric, requiring only the addition of two parameters to model objects with various shapes. The superquadric can be formally represented as the spherical product of two superellipsoids [26], described by the following parametric equations:

$$\mathbf{p}(\eta,\omega) = \begin{bmatrix} C_\eta^{\varepsilon_1} \\ a_z S_\eta^{\varepsilon_1} \end{bmatrix} \otimes \begin{bmatrix} a_x C_\omega^{\varepsilon_2} \\ a_y S_\omega^{\varepsilon_2} \end{bmatrix} = \begin{bmatrix} a_x C_\eta^{\varepsilon_1} C_\omega^{\varepsilon_2} \\ a_y C_\eta^{\varepsilon_1} S_\omega^{\varepsilon_2} \\ a_z S_\eta^{\varepsilon_1} \end{bmatrix} \tag{5}$$
$$C_\alpha^\varepsilon \triangleq sgn(\cos(\alpha))|\cos(\alpha)|^\varepsilon$$
$$S_\alpha^\varepsilon \triangleq sgn(\sin(\alpha))|\sin(\alpha)|^\varepsilon$$

where $\mathbf{p}(\eta,\omega)$ represents a point on the superquadric, $\otimes$ represents spherical product, and $-\pi/2 \le \eta \le \pi/2, -\pi \le \omega \le \pi$. The parameter a_i represents the axis length of the main axis, and ε_i controls the shape of the superquadric. To ensure that the object representation is a convex polyhedron and avoid optimization falling into local optimality, it is typically ensured that $0.1 < \varepsilon_i < 2.0$.

Algorithm 1 Expanding the Scope of Semantic Fine-tuning

Input: Original detections $\{O_{D_k}\}$ and $\{O_{D_{k+1}}\}$,Human correction detections $\{H_{D_k}\}$
Output: New correction detections $\{H_{D_{k+1}}\}$

```
if {H_{D_k}} is not null then
    for each D^i_{k+1} in {O_{D_{k+1}}} do
        Association{O_{D^i_{k+1}}, O_{D^j_k}} ← IoU(D^i_{k+1}, {O_{D_k}})
    end for
end if
//computer H_L
if Associations {{O_{D_{k+1}}}, {O'_{D_k}}} then
    H_L ← homography({O_{D_{k+1}}}, {O'_{D_k}})
end if
//computer {H_{D_{k+1}}}
H_{D^i_{k+1}} ← transform(H_L, H_{D^i_k})
Association{H_{D^i_{k+1}}, O_{D^j_{k+1}}} ← IoU(H_{D^i_{k+1}}, {O_{D_{k+1}}})
//correct label and confidence
if Association{H_{D^i_{k+1}}, O_{D^j_{k+1}}} is not null then
    (H_{D^i_{k+1}})_{label,box} ← (O_{D^j_{k+1}})_{label,box}
    (H_{D^i_{k+1}})_{con} ← ((H_{D^i_{k+1}})_{con} + (O_{D^j_{k+1}})_{con})/2
    Add H_{D^i_{k+1}} to {H_{D_{k+1}}}
else
    Add H_{D^i_{k+1}} to {H_{D_{k+1}}}
end if
return {H_{D_{k+1}}}
```

The superquadric can be expressed compactly with 11 parameters, denoted as $\Lambda = (\theta_1,\theta_2,\theta_3,t_1,t_2,t_3,a_x,a_y,a_z,\varepsilon_1,\varepsilon_2)$. For the convenience of parameter calculation, we independently estimate the parameter ε_i which controls the shape, while the initial values for the other nine parameters are referenced from [6], and directly obtained during the pose estimation in the tracking thread.

Map points will increase with the camera's movement. Once a certain number of map points have accumulated, projecting these map points onto the image frames allows obtaining a collection of map points within the detection box. However, at this stage, the map points may be mixed with background points and cannot be directly used to estimate shape parameters. To address this, the coordinates of the map points are sorted, and points with significantly large coordinate differences are excluded. Subsequently, the Isolation Forest algorithm in [7] is used to cluster the map points. The clustered map points are then processed using the convex hull construction method in [8] to obtain a minimal envelope. The position of the points on the envelope's surface relative to the superquadric surface is determined through an implicit function as follows:

$$F(x,y,z) = \left(\left(\frac{x}{a_x}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_y}\right)^{\frac{2}{\varepsilon_2}}\right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_z}\right)^{\frac{2}{\varepsilon_1}} = 1 \tag{6}$$

Then, it is only necessary to fit the best shape by minimizing the sum of squared Euclidean distances between points and the superquadric surface. Fig.4 illustrates the process of

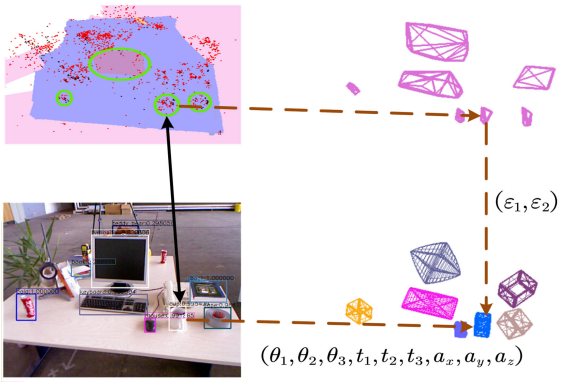


Fig. 4. **Initialization of Superquadric**. The pose parameters are obtained from semantic information and camera pose, while the shape is obtained by fitting the minimum convex hull constructed from the map points.

fitting the superquadric surface from map points.

$$\{\Lambda_c\}^{(opt)} = \underset{\{\Lambda_c\}}{arg\min} \sum_{{}^s\mathbf{p}\in{}^s\mathbb{P}} ||{}^c\mathbf{p}||^2 \left[H^{-\frac{1}{2}}\left({}^c\mathbf{p}, \Lambda_c\right) - 1\right]^2 \quad (7)$$

where $\Lambda_c = \{a_x, a_y, a_z, \varepsilon_1, \varepsilon_2\}$ is the parameter set for the superquadric and ${}^c\mathbf{p}$ represents a set of points on superquadric surface. We have already obtained a_i earlier, substituting them into Eq.7 to estimate ε_i. In Alg.2, we present the entire process of superquadric initialization.

D. Joint Optimization

We perform joint optimization of map points, map planes, camera poses, and superquadric landmarks in the local mapping thread. The comprehensive optimization function we have designed is articulated as follows:

$$\begin{aligned}
&\{T_{k,w}, Q_w, \pi_w, X_w\}^{\text{opt}} = \\
&\underset{\{T_{k,w}, Q_w^T\}}{arg\min} \sum\sum f_{\text{superquadric}}\left(T_{k,w}, Q_w^T\right) + \\
&\underset{\{Q_w^T, \pi_w\}}{arg\min} \sum f_{\text{tangent}}\left(\pi_w, Q_w^T\right) + \\
&\underset{\{Q_w^\varepsilon, X_w\}}{arg\min} \sum f_{\text{superquadric}}\left(X_w, Q_w^\varepsilon\right) + \\
&\underset{\{T_{k,w}, X_w\}}{arg\min} \sum\sum f_{\text{point}}\left(T_{k,w}, X_w\right) + \\
&\underset{\{T_{k,w}, \pi_w\}}{arg\min} \sum\sum f_{\text{plane}}\left(T_{k,w}, \pi_w\right)
\end{aligned} \quad (8)$$

where $T_{k,w}$ is the camera pose of the image frame I_k, and Q_w is the parameters of the superquadric. During optimization, we separate the shape control parameters Q_w^ε from the pose parameters Q_w^T. π_w represents the plane parameters, and X_w represents the map points. All these parameters are represented in the world coordinate system.

The constraint part between map points and the shape of superquadric in Eq.8 is consistent with Eq.7, while the other components are explained as follows:

- The constraint between superquadrics in the map and the detection boxes in frame I_k:

$$f_{\text{superquadric}}\left(T_{k,w}, Q_w^T\right) = ||D_k^{Q_w^T} - \eta\left(T_{k,w}, Q_w^T\right)||^2_{\Sigma_{sq}} \quad (9)$$

where $D_k^{Q_w^T}$ is the detection box associated with Q_w^T, and $\eta\left(T_{k,w}, Q_w^T\right)$ denotes the superquadric Q_w^T transformed into a 2D bounding box through the camera pose $T_{k,w}$.

Algorithm 2 Superquadric Initialization

Input: The corrected detection results $\{D_k\}$ in frame I_k and map points $\{X_w\}$
Output: The pose $\{T_w^Q\}$ and shapes $\{Q_w^{\Lambda_c}\}$ of new constructed superquadrics $\{\mathrm{Q}_w\}$

for each D_k^i in $\{D_k\}$ **do**
  //obtain multiple frames of observations
  Associations$\{D_k^i, D_{k-1}^j\} \leftarrow IoU(\mathrm{D}_k^i, \{D_{k-1}\})$
  Add D_k^i and D_{k-1}^j to $\{D_k\}$
  if $\{D_k\}$ and $\{X_w\}$ is sufficient **then**
    $\{\pi_w\} \leftarrow LineBackProjection(\{\mathrm{D}_k\})$
    //solving parameters using SVD
    $Q_w^p \leftarrow SolveSuperquadric(\{\pi_w\})$
    $T_w^Q, Q_w^{\Lambda_a} \leftarrow DecomposeParameters(\mathrm{Q}_w^p)$
    Add T_w^Q to $\{T_w^Q\}$
    //obtain the set of map points within the box
    $\{X_w^i\} \leftarrow MapPointsProjection(\{\mathrm{X}_w\}, D_k^i)$
    //map points filtering
    $\{X_w^i\} \leftarrow FilterByCoordinate(\{\mathrm{X}_w^i\})$
    $\{X_w^i\} \leftarrow FilterByDepth(\{\mathrm{X}_w^i\}, \{u_k^i\})$
    $\{X_w^i\} \leftarrow FilterByiForest(\{\mathrm{X}_w^i\})$
    $Q_w^{ch} \leftarrow BuildMinConvexHull(\{\mathrm{X}_w^i\})$
    $Q_w^{\Lambda_\varepsilon} \leftarrow FittingByEuclideanDistance(\mathrm{Q}_w^{ch})$
    Add $Q_w^{\Lambda_a}$ and $Q_w^{\Lambda_\varepsilon}$ to $\{Q_w^{\Lambda_c}\}$
  end if
end for
return $\{\{T_w^Q\}, \{Q_w^{\Lambda_c}\}\}$

- The superquadric and the plane are constrained using the tangent relation:

$$f_{\text{tangent}}\left(\pi_w, Q_w^T\right) = ||\pi_w^T Q_w^T \pi_w||^2_{\Sigma_t} \quad (10)$$

- The constraint between map points and feature points:

$$f_{\text{point}}\left(T_{k,w}, X_w\right) = ||T_{k,w} X_w - u_k^{(obs)}||^2_{\Sigma_{point}} \quad (11)$$

where u_k is the feature point coordinate in frame I_k.

- The constraint between map planes in the map and the planes extracted from frame I_k:

$$f_{\text{plane}}\left(T_{k,w}, \pi_w\right) = ||T_{k,w}^{-T} \pi_w - \pi_k^{(obs)}||^2_{\Sigma_{plane}} \quad (12)$$

where π_k is plane coefficients in frame I_k.

The joint optimization is performed through g2o.

V. Experiments

We conduct a performance evaluation of our proposed approach on the publicly available TUM RGB-D and Scenes V2 datasets, including the number and shape of objects reconstructed in the map and the accuracy of camera localization. All experiments are performed using an Intel(R) Core(TM) i7-11700H CPU @ 2.5GHz and 16GB RAM.

A. The Quality of Semantic Map

We use the number of correctly reconstructed objects in the map as the evaluation criterion. As current open-source projects still widely use quadrics to represent objects, we compare with the latest and influential work OA-SLAM

TABLE I
NUMBER OF OBJECTS CONSTRUCTED BY SUPERQUADRICS USING TUM RGB-D AND SCENES V2 SEQUENCES.

Dataset	Sequence	GT	OA-SLAM [20]	Ours	Ours+H
TUM RGB-D	fr1-desk	16	10	12	**13**
	fr1-desk2	17	9	11	**15**
	fr1-xyz	9	7	6	**8**
	fr2-desk	18	20	11	**16**
	fr3-office	22	41	15	**20**
Scenes V2	scene-01	5	3	3	**4**
	scene-04	5	4	4	**5**
	scene-07	7	4	3	**6**

Note: "GT" indicates the ground truth, and "H" indicates human-in-the-loop semantic refinement.

[20]. To validate our human-in-the-loop semantic refinement approach and ensure a fair comparison, we uniformly use the unmodified YOLOv5 to obtain detection results from images and compare the data on the same local platform. The experimental results are shown in Table I. We can observe that our system demonstrates a significant advantage in most sequences, especially after incorporating human-in-the-loop semantic refinement, where the number of correctly reconstructed objects in the map has increased.

In several sequences of the TUM dataset, our system outperforms OA-SLAM. OA-SLAM requires the use of quadrics for relocalization and does not undergo strict optimization in the system, relying instead on accurate object detection results. In sequences with more objects, such as fr2-desk and fr3-office, the untuned YOLOv5 algorithm we used has a higher rate of false detections, leading to duplicated constructions and misrepresentations in the maps it builds. With the incorporation of plane constraints in our system, effective constraints are imposed on the objects in the map. The data association component also limits the redundant representation of objects. Therefore, our system performs significantly better than OA-SLAM in these two sequences.

Additionally, in the two longer sequences, fr2-desk and fr3-office, there is no violent camera shake or fast angle change. In this case, it is convenient to calculate the local homography matrix between image frames, thus expanding the effectiveness of manual fine-tuning. This is also the reason our system performs exceptionally well in these two sequences. In contrast, the presence of rapid changes in the viewpoint in the fr1 sequence results in manual correction results that cannot be accurately associated with the next frame. Therefore, the performance improvement in the fr1 sequence is not as significant as in the other two sequences.

In the Scenes V2 dataset, we disregard objects like sofas that only appear for a short period and focus only on the statistical analysis of tabletop objects and adjacent chairs. In all three sequences, both OA-SLAM and our system successfully represent the objects. Furthermore, with the addition of manual fine-tuning, our system outperforms better.

B. The Quality of Superquadric

Superquadric can control shapes to represent different objects more precisely. Therefore, the representation of objects in the map becomes more detailed and distinguishable.

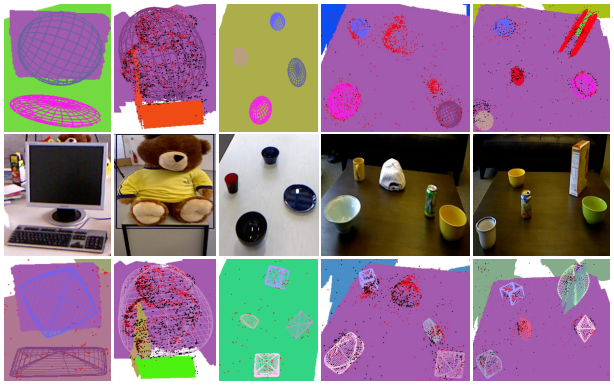


Fig. 5. **Object representation**. In the middle of each column are the RGB images of objects. Above are the quadric representations of the objects, and below are the representations using superquadrics.

Fig. 5 illustrates the representation of objects in different sequences of two datasets using quadrics and superquadrics. In addition to representing the position of objects in a map, superquadric can also represent the boundaries and orientations of objects. For example, a keyboard and a book represented by superquadric can directly distinguish the orientation of the objects. On the other hand, a monitor represented by an ellipsoid is challenging to differentiate from other objects with similar volumes. However, when represented by superquadric, it can be clearly distinguished in the map even without planar assistance.

C. The Accuracy of Localization

To validate the localization accuracy of our system, we conduct comparative analyses against ORB-SLAM2, SP-SLAM [17], OA-SLAM, and QuadricSLAM [6]. While the data for QuadricSLAM is extracted from previously published papers owing to not being open-source, data for the remaining systems are reproduced. The experimental findings are delineated in Table II.

Taking ORB-SLAM2 as the baseline standard and comparing the performance of these systems, QuadricSLAM has the poorest performance. This is because quadrics are primarily used to represent objects, and in terms of optimizing poses, only unoccluded quadrics can provide effective constraints. Due to observation limitations and the influence of system noise, the localization accuracy decreases significantly. OA-SLAM incorporates data association, allowing for more constraints during pose optimization. Therefore, compared with the baseline method, the localization accuracy of OA-SLAM shows moderate improvement.

Our work outperforms other comparison methods in sequences such as fr1-desk2, fr1-xyz, fr2-desk, and fr3-office. In addition to plane constraints, we correct semantic information using human-in-the-loop refinement, increasing the number of correctly represented objects in the map. This, in turn, allows for the construction of more constraints during pose optimization. However, in the fr1-desk sequence, due to rapid changes in the camera perspective and increased occlusion on the desk, it is challenging to segment effective planes. This results in our method not having a competitive advantage in terms of localization accuracy.

TABLE II
THE COMPARISON OF LOCALIZATION ACCURACY USING TUM RGB-D SEQUENCES.

Sequence	ORB-SLAM2 [1]	SP-SLAM [17]	QuadricSLAM [6]	OA-SLAM [20]	Ours	Ours+SQ	Ours+H
fr1-desk	0.0134	0.0144	0.0632	**0.0120**	0.0135	0.0131	0.0124
fr1-desk2	0.0207	0.0227	0.0662	0.1296	0.0186	0.0177	**0.0173**
fr1-xyz	0.0083	0.0082	-	0.0082	0.0083	0.0083	**0.0079**
fr2-desk	0.0106	0.0195	0.0568	0.0096	0.0075	0.0069	**0.0059**
fr3-office	0.0114	0.0156	0.0765	0.0107	0.0095	**0.0090**	0.0091
Average	0.0129	0.0161	0.0525	0.0340	0.0115	0.0110	**0.0105**

Note: "-" indicates that the data is not mentioned in the paper. "Ours" indicates that the plane is added by default, and "SQ" indicates that the superquadric is added.

VI. Conclusion

This paper introduces a novel human-in-the-loop object-level semantic SLAM called HSS-SLAM. This approach enables interaction between the user and the system, allowing for improvements to the semantic information in the global map through a few operations. We employ superquadric to alleviate the difficulty in distinguishing objects on the map. Extensive experiments on public datasets demonstrate that our proposed method outperforms other state-of-the-art methods. Future work will focus on integrating human-in-the-loop refinement with outdoor environments.

References

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[3] X. Zhang, W. Wang, X. Qi, and Z. Liao, "Stereo plane slam based on intersecting lines," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6566–6572.

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.

[5] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[6] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018.

[7] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, and D. Kerr, "Eao-slam: Monocular semi-dense object slam based on ensemble data association," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4966–4973.

[8] Z. Cao, Y. Zhang, R. Tian, R. Ma, X. Hu, S. Coleman, and D. Kerr, "Object-aware slam based on efficient quadric initialization and joint data association," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9802–9809, 2022.

[9] Y. Wang, B. Xu, W. Fan, and C. Xiang, "Qiso-slam: Object-oriented slam using dual quadrics as landmarks based on instance segmentation," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2253–2260, 2023.

[10] S. Nashed and J. Biswas, "Human-in-the-loop slam," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[11] A. Sidaoui, M. K. Zein, I. H. Elhajj, and D. Asmar, "A-slam: Human in-the-loop augmented slam," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 5245–5251.

[12] K. Ok, K. Liu, K. Frey, J. P. How, and N. Roy, "Robust object-based slam for high-speed autonomous navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 669–675.

[13] S. Chen, S. Song, J. Zhao, T. Feng, C. Ye, L. Xiong, and D. Li, "Robust dual quadric initialization for forward-translating camera movements," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4712–4719, 2021.

[14] R. Tian, Y. Zhang, Y. Feng, L. Yang, Z. Cao, S. Coleman, and D. Kerr, "Accurate and robust object slam with 3d quadric landmark reconstruction in outdoor environment," in *2022 IEEE International Conference on Robotics and Automation*, 2022.

[15] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang, and W. Wang, "So-slam: Semantic object slam with scale proportional and symmetrical texture constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4008–4015, 2022.

[16] Z. Fang, J. Han, and W. Wang, "Detect orientation of symmetric objects from monocular camera to enhance landmark estimations in object slam," *Applied Sciences*, vol. 13, no. 4, p. 2096, 2023.

[17] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-plane slam using supposed planes for indoor environments," *Sensors*, vol. 19, no. 17, p. 3795, 2019.

[18] Z. Liao, W. Wang, X. Qi, and X. Zhang, "Rgb-d object slam using quadrics for indoor environments," *Sensors*, vol. 20, no. 18, p. 5150, 2020.

[19] F. Shu, J. Wang, A. Pagani, and D. Stricker, "Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2105–2112.

[20] M. Zins, G. Simon, and M.-O. Berger, "Oa-slam: Leveraging objects for camera relocalization in visual slam," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 720–728.

[21] A. Makhal, F. Thomas, and A. P. Gracia, "Grasping unknown objects in clutter by superquadric representation," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 292–299.

[22] A. Gomez, S. Rilling, and R. Herpers, "Superquadric indoor scene representation for orientation and navigation tasks," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 3692–3698.

[23] D. Paschalidou, A. O. Ulusoy, and A. Geiger, "Superquadrics revisited: Learning 3d shape parsing beyond cuboids," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 336–10 345.

[24] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal, "Human-in-the-loop machine learning: A state of the art," *Artificial Intelligence Review*, vol. 56, no. 4, pp. 3005–3054, 2023.

[25] Z. Ouyang, C. Zhang, and J. Cui, "Semantic slam for mobile robot with human-in-the-loop," in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer, 2022, pp. 289–305.

[26] N. Vaskevicius and A. Birk, "Revisiting superquadric fitting: A numerically stable formulation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 220–233, 2017.