SQ-SLAM: Monocular Semantic SLAM Based on Superquadric Object Representation

Xiao Han · Lu Yang

Received: 1 March 2023 / Accepted: 29 August 2023 / Published online: 22 September 2023
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

Abstract
Object SLAM uses additional semantic information to detect and map objects in the scene, in order to improve the system’s perception and map representation capabilities. Previous methods often use quadrics and cuboids to represent objects, especially in monocular systems. However, their simplistic shapes are insufficient for effectively representing various types of objects, leading to a limitation in the accuracy of object maps and consequently impacting downstream task performance. In this paper, we propose a novel approach for representing objects in monocular SLAM using superquadrics (SQ) with shape parameters. Our method utilizes object appearance and geometry information comprehensively, enabling accurate estimation of object poses and adaptation to various object shapes. Additionally, we propose a lightweight data association strategy to accurately associate semantic observations across multiple views with object landmarks. We implement a monocular semantic SLAM system with real-time performance and conduct comprehensive experiments on public datasets. The results show that our method is able to build accurate object maps and outperforms state-of-the-art methods on object representation.

Keywords
Visual SLAM · Semantic mapping · Object representation · Superquadrics

1 Introduction

Visual simultaneous localization and mapping (SLAM) is one of the basic technologies of robot spatial perception, which has been widely used in mobile robots, autonomous driving and AR/VR. Traditional methods use geometric primitives such as points [1], lines [2] and planes [3] in the scene to build maps and simultaneously estimate the pose of sensors. Their sparse, semi-dense or dense metric maps focus on building an accurate geometric representation of the world. However, maps containing only pure geometry information limit high-level tasks. For example, mobile robots need to perceive semantic information for task-oriented navigation [4] and in AR applications, 3D objects need to be localized to achieve virtual interaction. The rapid development of deep learning in recent years has paved the way for further enhancing the robot’s perception and map representation capabilities. The robotics community is increasingly introducing learning-based methods [5, 6] into SLAM systems.

Object SLAM is one of the challenging applications that combines semantic information with visual SLAM. By incorporating objects as map elements during the observation process, the robot’s scene understanding [7, 8] is enhanced, thereby facilitating the execution of more intricate tasks [9]. On the other hand, the observation of some long-term consistent objects has a positive effect on the long term operation and relocalization of the SLAM system [10, 11]. Early work [12] used object CAD models and point cloud information to construct object maps. However, the reliance on prior information and additional sensors restricted its applicability in various scenarios. In recent work, it is popular to use cubes [13, 14] or ellipsoids [15–17] to represent objects, and object map is constructed by only 2D bounding boxes without prior 3D models. This compact geometric representation retains basic space occupation information of objects. However, simple cubes or ellipsoids cannot effectively represent the shapes of various objects, potentially leading to inaccurate scene understanding or affect object interaction.

In this work, we propose SQ-SLAM, a monocular semantic SLAM based on superquadric object representation,
which constructs an object-oriented map as shown in Fig. 1. Superquadrics are extensions of standard quadrics, which can represent common 3D geometric primitives such as ellipsoids, cylinders and cubes. We introduce convex superquadrics with zero pitch and roll angles into object SLAM, enabling the accurate representation of different types of objects using a set of abstract geometric parameters that include pose, size and shape. Due to the computational complexity associated with directly optimizing all parameters, we decouple the pose and shape parameters and estimate them independently within the front-end and back-end threads of the SLAM system, thereby guaranteeing real-time performance. In addition, accurately associating multi-view observations at different times to object landmarks presents another challenge in object SLAM. We propose a lightweight data association strategy that comprises consecutive association and non-consecutive association. The former uses object appearance information such as 2D bounding boxes and inter-frame feature matching, while the latter utilizes parametric statistical tests to evaluate the geometric relationship between objects. Comprehensive experiments on public datasets demonstrate that our method outperforms baselines in terms of object representation strength.

The contributions of this work are as follows:

- We propose a superquadric object representation method with separate parameter estimation, enabling the construction of semantically enhanced object maps.
- We leverage the appearance information of objects and the geometry information of the sparse point cloud generated by triangulation during camera tracking to implement a lightweight data association strategy.
- Based on the proposed algorithm, we implement a real-time monocular object SLAM, and verify its effectiveness on public datasets.

2 Related Work

2.1 Object Representation in SLAM

There are various object representations in many literatures on semantic SLAM, such as prior object models, point cloud clusters with semantic labels, 3D bounding boxes, etc. Salas et al. [12] were the first to introduce objects as landmarks into SLAM system. They used RGB-D information and prior object models to construct maps, but it cannot be applied to general scenes. Maskfusion [18] accurately constructs dense object models with the help of semantic segmentation, which are represented as surfel clouds. The system does not need prior models, but consumes a lot of computational resources. In contrast, Sunderhauf et al. [19] avoid pixel-by-pixel processing and perform unsupervised segmentation only on dense point clouds located in detection bounding boxes. Point cloud clusters with semantic labels are used to represent objects, but they lack spatial metric attributes.

Implementing object SLAM using only monocular camera has also garnered significant interests. In 2019, Yang et al. proposed CubeSLAM [13], which uses 3D cubes inferred from 2D bounding boxes as object representation. Compared with dense models, this simple geometric primitive retains the basic space occupation information of objects, such as position, rotation and scale. In addition, quadrics are favored by researchers due to their simple mathematical representation and compact perspective projection model. Rubino et al. [20] proposed an analytical method for multi-view localization of objects. QuadricSLAM [17] introduces quadrics into SLAM as object landmarks, and optimizes their parameters through continuous image observation after initialization. Recently, researchers have focused on improving the accuracy and robustness of quadric landmarks. For instance, Tian et al. [21] proposed a parameter-separated initialization method to make it suitable for vehicles in outdoor environments. SO-SLAM [22] explores the symmetry of objects and implements an orientation fine-tuning algorithm. Hu et al. [23] introduced symmetric positive-definite matrix manifold to solve the singularity issue of classical quadric parameterization method.

However, the single geometric model used in the aforementioned methods is difficult to fit for various types of
objects. Kimera [24] estimates 3D bounding boxes for objects of unknown shape and uses CAD models to fit known objects. Zhen et al. [25] used quadrics and their degenerate cases to uniformly represent geometric primitives like planes, ellipsoids and cylinders. Tschopp et al. [26] explored superquadrics, running instance segmentation on multi-view images and then retrieving the parameters of superquadrics using the mask contours. But they only verified it on preliminary simulation experiments, and did not build a complete object SLAM system. In contrast, our proposed method does not require additional semantic segmentation, which utilizes sparse point clouds incrementally generated during the SLAM process to model superquadrics.

2.2 Data Association

Data association is another widely studied problem in semantic SLAM. Its goal is to correctly associate multiple object observations at different positions and times with the same object landmark. Bowman et al. [27] proposed an expectation maximization (EM) algorithm for soft data association, and tightly coupled metric information, semantic information and data association into a unified optimization framework. Doherty et al. [28] proposed the max-marginalization procedure and semantic max-mixture factors for optimization. However, these algorithms based on EM or max-marginalization do not consider the rich information contained in the appearance of objects.

Data associations can also be solved directly by object observations. Yang et al. [13] associated objects with feature points located in 2D detection boxes, and then inferred object associations through inter-frame feature matching. Similarly, [29] calculates the Bag of Words (BoW) vector of feature points as the appearance description of objects, and then perform BoW matching on the candidate objects that satisfy the reprojection relationship. Chen et al. [30] proposed a hierarchical object association strategy, and applied multi-object tracking for short-term object association. Although these methods based on object appearance are efficient, they are sensitive to observation noise and not robust enough. Iqbal et al. [31] used nonparametric statistical test to address the non-gaussian property of object point cloud distributions. [14] extends it, taking into account the statistical properties of the point cloud centroid. Inspired by [14], our work leverages the appearance and geometry information of objects, and the implemented association algorithm is lightweight to ensure real-time performance.

3 Superquadrics

Superquadrics are extensions of standard quadrics, first proposed by Barr et al. [32]. They are a series of parametric surfaces that include superellipsoids, supertoroids, and super-hyperboloids with one piece and two pieces. In this paper, we focus on superellipsoids, and follow community convention to refer to them by the more generic term of superquadrics. A superquadric can be obtained by a spherical product of two superellipses, $s_1$ and $s_2$, which is described by the following parametric equation:

$$
p(\eta, \omega) = s_1(\eta) \otimes s_2(\omega) = \begin{bmatrix} a_x \cos^{\epsilon_1} \eta \cos^{\epsilon_2} \omega \\ a_y \cos^{\epsilon_1} \eta \sin^{\epsilon_2} \omega \\ a_z \sin^{\epsilon_1} \eta \end{bmatrix}
$$

(1)

where $p(\eta, \omega)$ denotes a point on the superquadric surface determined by two angle variables $\left( -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2}, -\pi \leq \omega \leq \pi \right)$. The parameters $a_x$, $a_y$ and $a_z$ define the size of the superquadric in three directions respectively, while $\epsilon_1$ and $\epsilon_2$ control the shape. Since the shapes of most common objects are convex, and in order to ensure numerical stability during the optimization process [26, 33], we limit the value range of shape parameters, i.e. $0.1 \leq \epsilon_1, \epsilon_2 \leq 1.9$. The shapes corresponding to different $\epsilon_1$ and $\epsilon_2$ are shown in Fig. 2.

When $\epsilon_1 = \epsilon_2 = 1$, it degenerates into a standard ellipsoid. Superquadrics can also be expressed by the following implicit equation:

$$F(x, y, z) = \left( \frac{x}{a_x} \right)^{2/\epsilon_1} + \left( \frac{y}{a_y} \right)^{2/\epsilon_1} + \left( \frac{z}{a_z} \right)^{2/\epsilon_1} = 1
$$

(2)

This function is also called inside-outside function, because it can be used to determine where the point lies with respect to the superquadric surface. If $F = 1$, the given point is located on the surface.

Defining a superquadric in the world coordinate system requires determining its rigid motion relative to the origin, including $t_x$, $t_y$, $t_z$ for the translation and $\rho$, $\psi$, $\theta$ for determining the rotation. A total of 11 parameters can represent a superquadric in general position as

$$\Lambda = \{a_x, a_y, a_z, \epsilon_1, \epsilon_2, \rho, \psi, \theta, t_x, t_y, t_z\}
$$

(3)

For more properties of superquadrics, please refer to [34]. Compared to other commonly used parameterized geometric primitives, such as cubes, ellipsoids and cylinders, superquadrics only require the addition of two shape parameters to elevate these geometric representations to the same parameter space. This enables the utilization of a consistent optimization method for fitting diverse shapes, eliminating the need for pre-identifying the shape category. Moreover, although some learning-based methods have the capability to capture the intricate shape and texture of objects, they are typically limited to predefined categories and require a large amount of parallel computing resources. In contrast, our
method achieves a favorable balance between performance and efficiency by enhancing the representation capabilities of objects with only a slight increase in computational cost.

4 SQ-SLAM

SQ-SLAM is built upon the feature-based ORB-SLAM2. Figure 3 illustrates the system’s components and data flow, with yellow-colored modules indicating pre-existing components. For a calibrated monocular video stream input, we first use YOLO [5] to detect objects from RGB images, resulting in 2D bounding boxes and class labels. These observations are then incorporated into the metric SLAM system as semantic information. After associating the latest object observations with landmarks, how to efficiently estimate the parameters of superquadrics is the main challenge of this work.

In the tracking thread, we perform camera pose estimation and triangulation of matched image features concurrently to generate a sparse point cloud. Compared with some works [35, 36] that commonly use additional depth sensors to retrieve superquadric parameters from dense point clouds, it is difficult to estimate all parameters directly from the sparse point cloud. Therefore, we decouple the object’s pose and shape and estimate them separately in two parallel threads. The tracking thread only estimates the pose of objects to ensure precise data association and real-time performance. Meanwhile, the mapping thread processes keyframes at a lower frequency, enabling ample time for estimating object shapes and constructing object maps with semantic class information.

4.1 Outlier Removal

The point cloud map generated by feature point-based SLAM is usually sparse. We begin by associating objects with their corresponding map points. In the tracking thread, the feature points within bounding boxes extracted in each frame will be associated with objects. However, since objects are always located in the background rather than isolated, and being susceptible to occlusion and measurement noise, this process unavoidably incorporates many outliers that do not belong to objects. We primarily employ the following two methods to remove outliers.

4.1.1 Reprojection

It is expected that the points obtained by reprojecting the object’s map points onto images from different viewpoints should all fall within the bounding boxes. Once an object is detected in the current frame, each map point belonging to the object is reprojected onto the image using the frame pose and intrinsic parameters. If the projection point falls outside the bounding box, the corresponding map point will be removed as depicted in Fig. 4a. Nevertheless, this method can solely remove outliers resulting from past observations at various viewpoints, and it lacks the capability to address new outliers introduced by the current frame, leading to a delay. To enhance robustness, we integrate it with the subsequent anomaly detection-based method.
Fig. 3  Overview of the system. The yellow blocks symbolize pre-built modules. By leveraging additional 2D semantic observations, we utilize superquadrics to model objects and estimate their pose and shape in two parallel threads. The resulting object map has both metric and semantic information.

Fig. 4  Two methods of outlier removal. (a) When generating observation images $I_k$ from different viewpoints, those reprojection points outside the bounding box are removed. (b) The green point needs 8 steps to be completely isolated. In contrast, the red point only needs 4 steps, so the latter is more likely to be an outlier.
4.1.2 Extended Isolation Forest

The points on the object surface tend to be densely distributed due to continuous multi-frame observations, with outliers being relatively rare and isolated. Based on this insight, we employ Extended Isolation Forest (EIF) [37] to tackle this issue. EIF recursively partitions the sample space until every data point is isolated, while outliers require fewer steps. Figure 4b illustrates this process. EIF partitions the data using hyperplanes with random slopes, which offers greater reliability compared to the standard version that employs hyperplanes parallel to the coordinate axes. Additionally, EIF effectively eliminates the artifact problem present in the standard version. The artifact problem arises due to the inaccurate estimation of the anomaly probability distribution caused by exclusively utilizing hyperplanes along the dimensional directions.

The algorithm first constructs a binary isolation tree. For a three dimensional point cloud \( X \), the plane normal vector \( n \) for each branch cut is easily obtained by sampling a standard three dimension gaussian distribution \( \mathcal{N}(0, 1) \). Due to the isotropy of the standard multivariate gaussian distribution, the sampled normal vectors are uniformly distributed. Then the branching criteria for a given point \( x \in X \) is as follows:

\[
x \cdot n \leq p
\]

where \( p \) is a random intercept that lies in the range of the branch data. If this condition is met, the point is assigned to the left branch, otherwise it is assigned to the right branch. We recursively perform the above process until each point is isolated or the depth limit is reached. By creating many such trees, we use the average depth of each point to evaluate its anomaly score:

\[
s(x, n) = \frac{2}{E(h(x))} E(h(x))
\]

\[
c(n) = 2H_{n-1} - \frac{2(n-1)}{n}
\]

\[5\]

\[6\]

Fig. 5 For visualization, the object frame is moved from the center to the vertices. (a) The line segments are projected onto the image. (b) Optimizing the error between projected and detected line segments to estimate yaw angle.

where \( E(h(x)) \) is the average depth of point \( x \) after traversing all trees, \( c(n) \) is a normalization factor, and \( H \) is a harmonic number. We use EIF when detecting new observations or merging object landmarks, and remove those points whose anomaly scores greater than a set threshold to maintain a sparse point cloud that accurately represents objects.

4.2 Pose Estimation

In the tracking thread, we estimate object pose using image observations and sparse point clouds. Generally, objects conform to gravitational effects and tend to align parallel to their supporting surface [22]. We make the assumption that the pitch and roll angles of objects remain constant at zero, requiring estimation solely for the yaw \( \theta \) and translation \( t \in \mathbb{R}^3 \). The pose of objects is represented by \( T_o = [R(\theta) \ t] \). Where \( R(\theta) \) is rotation matrix. Initially, we calculate the centroid of the sparse point cloud as a straightforward method to estimate \( t \). Subsequently, we employ various methods to estimate rotation based on the object’s appearance.

In the case of objects with well-defined edges like books, chairs and keyboards, the ideal alignment of the modeled superquadric’s coordinate axis is parallel to the object edges. We first use EDLines [38] to extract line segments from the image and associate those located within the bounding boxes with the corresponding objects. If the count of associated lines is greater than a threshold, as shown in Fig. 5, the unit line segments \( l_i (i \in \{1, 2, 3\}) \) on the three coordinate axes of the superquadric are projected onto the image respectively. Finally, the accumulative angle error between the projected line segments \( l_{oi} \) and the detected line segments \( l_{ed} \) is optimized to estimate object yaw \( \theta \). The error function is defined as follows:

\[
\theta^* = \arg \min_{\theta} \sum_{i=1}^{3} \| g(l_{oi}) - g(l_{ed}) \|^2
\]

\[
l_{oi} = K T_{e}^{-1} (R(\theta) l_i + t), \ i \in \{1, 2, 3\}
\]

\[7\]

\[8\]
where \( g(\cdot) \) calculates the slope of line segment, and \( l_{ci,ed} \) denotes the detected line segment that matches \( l_{si} \) (angle error \(< 5^\circ\) ). \( K \) refers to the camera intrinsic matrix, and \( T_c \) represents the camera pose. Since optimizing rotation is a nonlinear process, obtaining a suitable initial value is crucial. We uniformly sample eighteen angles from \(-45^\circ \) to \(45^\circ\), and count the number of matched line segments after projection. The sample with the highest count of matches is utilized as the initial value for optimization.

In the case of objects like bowls and cups where straight edges cannot be detected, we employ principal component analysis (PCA) to determine their orientation. Since only the yaw angle is estimated, we project the sparse point cloud of the object onto the X-Y plane, and then directly calculate its rotation matrix corresponding to the dominant direction using PCA. It is important to note the presence of a singular rotation matrix problem [23], i.e. rotating the superquadric coordinate frame by \(90^\circ\) and swapping the scale values of two directions, the result represents the same object. Hu et al. [23] introduced symmetric positive-definite matrix manifold to solve this problem, but they dealt with standard quadrics. We use a more concise approach to correct the results of PCA by automatically rotating objects by \(90^\circ\), \(-90^\circ\) and \(180^\circ\) degrees. The angle between the \(X\) axis of object frame and the \(X\) axis of world frame is always less than \(90^\circ\), so as to ensure that the object pose is globally consistent. Finally, we continuously update the object rotation as follows:

\[
\theta_n = \left(1 - \frac{1}{n}\right)\theta_{n-1} + \frac{1}{n}\theta_n^* \tag{9}
\]

where \( n \) is the number of observations. With the increase of \( n \), the object pose is gradually stabilized.

### 4.3 Size and Shape

After estimating superquadric pose in the tracking thread, we retrieve the remaining parameters in the mapping thread. First, a local object frame is assumed at the object center, and the sparse point cloud \( P^O \) in object frame is obtained by using the object pose \( T_o \). Under the influence of the robust outlier Removal algorithm, point cloud can accurately approximate the space occupation of objects, so we directly calculate the superquadric size \( a = [a_x, a_y, a_z]^T \) as follows:

\[
a = \frac{\max(P^O) - \min(P^O)}{2} \tag{10}
\]

Compared to Eq. (2), a better performance objective function [39] based on radial euclidean distance is used to fit superquadric shape, which is defined as the distance from any point to the surface of the superquadric \( A \):

\[
G(A, P^O) = \left\| P^O \right\| \left| 1 - F^{-\frac{1}{2}}(P^O) \right| \tag{11}
\]

Using all map points of the object, the optimization function is as follows:

\[
\{\varepsilon_1, \varepsilon_2\}^* = \arg \min_{\{\varepsilon_1, \varepsilon_2 \in A\}} \sum_{i=1}^{p^O} G(A, P^O)^2_{\Sigma_G} \tag{12}
\]

\[
\alpha_i = \frac{\left\| P^O_i \right\| - \min \left\| P^O \right\|}{\max \left\| P^O \right\| - \min \left\| P^O \right\|} \tag{13}
\]

where \( \Sigma_G \) is a diagonal covariance matrix, which is determined by the uncertainty \( \alpha \) of all shape errors. We use Levenberg-Marquardt algorithm to solve the above optimization problem.

Please note that it is not appropriate to directly minimize the radial euclidean distance of all points. On the one hand, many points of non-convex objects such as sofas and chairs are located near the center of their abstract models, which is not suitable for directly fitting superquadrics. On the other hand, due to the influence of observation noise and pose estimation errors, many points are not located near the surface, but are scattered inside objects. Therefore, we assign different optimization weights to errors according to the distance from the point to the center of superquadric. Furthermore, some works [13, 14] jointly optimize objects, map points, and camera poses. However, we found that it is more accurate and robust to fix camera pose and map points and only optimize the object parameters. Due to the non-convex nature of some objects and observation noise, some errors constructed by Eq. (2) are not realistic, and the misleading gradients of these errors actually deteriorate the camera pose estimation. We therefore offload the camera pose estimation to conventional reprojection errors, whose performance is consistent with the original ORB-SLAM2. Then, the fixed map points are used to estimate object parameters. Similar to rotation, optimizing the above nonlinear process also requires a good initial value to avoid converging into local minima. We sample in the parameter space \([0.1, 1.9] \times [0.1, 1.9] \) and take the sample with the smallest distance error as the initial value of optimization.

### 4.4 Data Association

The above algorithm continuously optimizes superquadric parameters in the SLAM framework, on the premise that object observations at different times and places can be accurately associated with landmarks. Our proposed lightweight
data association strategy using object detection information and sparse point clouds. It consists of the following two parts.

### 4.4.1 Consecutive Association

Since the change of camera pose between consecutive frames is small, the bounding boxes of the same object have a large overlap in consecutive images. The commonly used intersection over union (IoU) method is considered first. For the semantic measurement \( z_i = \{b_i, c_i\} \) with bounding box \( b_i \) and class label \( c_i \), we traverse the object landmarks with the same label \( c_i \) that were successfully associated in the previous frame, and its association is determined by the IoU of the two bounding boxes. However, this method is fragile in the case of multiple similar objects. We use inter-frame point association as a complement to the IoU method. Part of the features extracted from the same object in consecutive frames will be associated with the same map points, and this point association based on feature descriptor matching is more robust. Therefore, the shared number of the points associated with the semantic measurement and the landmark’s point cloud can also be used to determine object association. We combine these two methods to improve the applicability of consecutive association.

### 4.4.2 Non-Consecutive Association

Affected by object occlusion and viewpoint changes, object detector cannot continuously output semantic measurements in each frame. Since the object centroids observed in different frames usually follow a gaussian distribution [14], we use the single-sample t-test to deal with isolated semantic measurements in time series. There are current measurement \( z_i \) and an object landmark \( O = \{p_1^O, \ldots, p_n^O\} \), where \( p^O \) are the historical observations of centroid. First calculate the centroid \( \bar{p}^O \) of the map points associated with \( z_i \), and suppose the null hypothesis is that \( z_i \) is an observation of the landmark \( O \), then the t statistic is defined as follows:

\[
t = \frac{\sqrt{n} (\bar{O} - \bar{p}^O)}{\sigma_O} \sim t (n - 1)
\]  

(14)

Given a significance level \( \alpha \), the critical value from the t-distribution on \( n - 1 \) degrees of freedom is \( t_{\alpha, n-1} \). If \( |t| \leq t_{\alpha, n-1} \), accept the null hypothesis and associate the semantic measure \( z_i \) with the landmark \( O \).

The above algorithm performs data association in real time. However, due to object detector errors or large viewpoint changes, the same object will correspond to repeated landmarks. We use the double-sample t-test to merge object landmarks. Suppose the null hypothesis is that the historical observations of two landmarks \( O_1 \) and \( O_2 \) are from the same object, then the t statistic is defined as follows:

\[
t = \frac{\bar{O}_1 - \bar{O}_2}{\sigma_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t (n_1 + n_2 - 2)
\]  

(15)

\[
\sigma_p = \sqrt{\frac{(n_1 - 1) \sigma_{O_1}^2 + (n_2 - 1) \sigma_{O_2}^2}{n_1 + n_2 - 2}}
\]  

(16)

where \( \sigma_p \) is the pooled standard deviation. Similarly, if \( |t| \leq t_{\alpha, n_1+n_2-2} \), the null hypothesis is accepted. We merge the two object landmarks and re-estimate superquadric parameters.

## 5 Experiments

We evaluate the effectiveness of our proposed method through comprehensive experiments conducted on three publicly available datasets: ICL-NUIM [40], TUM RGB-D [41], and RGB-D Scenes v2 [42]. We also compare our results with those achieved by other state-of-the-art approaches. These datasets cover a wide range of indoor scenes, such as living rooms, offices, and desktops, which contain various types of objects that can be detected and mapped as semantic landmarks.

The 3D intersection over union (IoU) is employed as an evaluation metric to gauge the quality of object mapping, providing a comprehensive measure of both object pose and shape. To establish the ground truth for objects, we manually label them in the globally dense point cloud provided by the datasets, assigning distinct geometric models (e.g., cubes, ellipsoids, cylinders) based on their shapes. Each sequence is executed three times, and the average is computed as the final result to minimize the impact of multithreading. Additionally, considering the absence of absolute scale in monocular SLAM, we scale the object map by aligning the camera poses. Regarding the EIF algorithm, we generate a fixed number of 30 trees and establish the anomaly score threshold for outliers as 0.6.

### 5.1 Shape Fitting

To assess the benefits of utilizing superquadrics as object representations, we initially evaluate the quality of landmarks produced by individual objects. We choose representative objects from the TUM \texttt{fr3_office}, ICL-NUIM \texttt{living_room_2}, and RGB-D Scenes v2 \texttt{07} and \texttt{09} sequences, which exhibit diverse shapes. Furthermore, we set the shape parameters of the superquadric to 1.0 to degenerate it into standard quadrics, allowing for a comparison between the two with identical pose and size.
Fig. 6 Visualization of different objects, where blue and green grids represent quadric and superquadrics respectively

Table 1 Comparison of 3D IoU for different object representations

| Objects | box  | book | vase | bowl | sofa |
|---------|------|------|------|------|------|
| Quadrics | 0.512 | 0.528 | 0.482 | 0.636 | 0.548 |
| Superquadrics | **0.631** | **0.622** | **0.642** | **0.644** | **0.756** |

The best results are highlighted in bold.

The visualization results are shown in Fig. 6, illustrating the superimposition of the system’s object landmarks onto the dense point cloud or mesh. The quadrics and superquadrics are represented by blue and green grids, respectively. It is evident that while the size is relatively accurate, the quadric shape significantly deviates from the actual object and fails to effectively represent the spatial occupancy information. In contrast, superquadrics exhibit a better ability to conform to the object’s shape, adapting well to different object types and providing higher object representation capability.

Table 1 presents the quantitative results. Superquadrics outperform quadrics in representing the object edges of box, book, sofa, and other similar cubic objects, resulting in an IoU increase of **11.9%**, **9.4%** and **20.8%**, respectively. In the case of a bowl, where the surface is smooth and its shape closely resembles an ellipsoid, the advantage of using superquadrics is less apparent.

5.2 Object Map

Subsequently, we assess the quality of the semantic object map constructed in complete sequences and compare it with the state-of-the-art methods QuadricSLAM [17] and SO-SLAM [22], utilizing the results provided in their papers. In contrast to our system’s employment of an automatic data association algorithm, these two methods rely on manual association of object observations across different frames. Additionally, the number of objects evaluated is limited compared to the total number of objects observable in the images. Consequently, we exclude objects with few observations or located far from viewpoints. The quantitative results are provided in Table 2. Similarly to Section A, we denote the results of standard quadrics as OursQ for further comparison, while the results obtained using superquadrics are labeled as OursSQ. For a more intuitive visualization, please refer to the supplementary video.

5.2.1 ICL-NUIM Dataset

The room2 sequence captures a living room scene, including large objects such as sofas and chairs. Figure 7a shows the visualization results. Our method can generate semantic object landmarks with precise pose estimation, and the superquadrics effectively conform to the object shapes. Due to the smooth camera motion and absence of object occlusion, the results of oursQ closely resemble those of SO-SLAM. In contrast, oursSQ demonstrates a substantial improvement in landmark quality, resulting in a **10.9%** increase in IoU. This is attributed to the presence of numerous objects in the sequence that deviate significantly from ellipsoids, showcasing the advantageous performance of superquadrics.

5.2.2 TUM RGBD Dataset

The three sequences capture distinct desktop scenes, with the camera orbiting the observed objects. Some of the objects are cluttered and occluded, as shown in Figs. 1 and 7b. This poses a challenge for object landmark estimation. Quadric and SO-SLAM directly use bounding box measurements to retrieve landmark parameters through the quadric projection.
model, which is susceptible to measurement noise and object occlusion. Especially for fr1_desk, these methods do not perform very well due to the fast camera movement and the interference of observations from multiple similar objects. In contrast, our outlier removal algorithm can effectively handle object occlusion through continuous viewpoint changes and is robust to object detection noise. Quantitative results indicate that our algorithm surpasses the comparison methods in constructing high-quality object maps, with the use of superquadrics representation further enhancing the 3D IoU. Fr2_dishes is an exception, as it contains only ellipsoid-like objects like bowls and plates.

5.2.3 RGBD Sences v2 Dataset

The dataset comprises of 14 different tabletop scenes that contain many common small objects such as boxes, bowls and cups. The objects of interest are positioned on the table, while the camera moves around the table. The quantitative results are shown in the bottom four rows of Table 2. As certain objects fall outside the scope of object detection, we evaluate our method on partial sequences, as depicted in Fig. 7c and d. Despite the presence of textureless objects like bowls, our algorithm can accurately estimate the pose and shape by utilizing edge points of the objects. Consistent with the previous experiments, the incorporation of superquadrics representation enhances the quality of object maps, validating the effectiveness of our method.

5.3 Data Association

We validate the performance of our proposed lightweight data association strategy by comparing it with the methods presented in [31] and EAO-SLAM [14]. The results from their respective papers are utilized for comparison. The number of object landmarks in the constructed map serves as a quantitative evaluation metric, and the results are presented in Table 3, where GT represents the number of ground truth objects. In several sequences of the Scenes v2 dataset, our point cloud-based non-consecutive association algorithm
Table 3  Quantitative analysis of data association

| Seq     | [31] | EAO-SLAM [14] | Ours | GT |
|---------|------|---------------|------|----|
| TUM     |      |               |      |    |
| fr1_desk| 14   | 14            | 14   | 16 |
| fr2_desk| 11   | 22            | 22   | 26 |
| fr3_office| 15     | 42            | 38   | 45 |
| fr3_teddy| 2     | 6             | 6    | 7  |
| Sences v2| 01  | 5             | 7    | 8  |
|         |      |               |      |    |
|         | 07   | 7             | 7    | 7  |
|         | 10   | 6             | 7    | 6  |
|         | 13   | 3             | 3    | 4  |
|         | 14   | 4             | 5    | 6  |

The metric is the number of objects associated successfully
The best results are highlighted in bold

readily achieves accurate association owing to the limited number of objects and their dispersed locations. The number of generated objects is very close to GT. Conversely, the TUM dataset presents more challenges, as exemplified by fr3_offce depicted in Fig. 7b. This particular scene contains numerous clustered bottles, which are susceptible to errors during the association process. The clustering methods employed in [31] are insufficient for addressing this challenge, whereas our method excels in accurately associating a majority of objects and performs comparably to EAO-SLAM. Moreover, in terms of efficiency, our lightweight strategy outperforms its ensemble method.

5.4 Ablation Study

5.4.1 Outlier Removal

We conduct an ablation study to demonstrate the effectiveness of integrating the two approaches for outlier removal. We additionally annotate the true size of the object during system operation, and judge whether the remaining points are within the true size after each execution of outlier removal algorithm, calculating the ratio of outliers to inliers. The mean of all observations serves as the evaluation metric, with lower values indicating better performance. The results on the Sense v2 07 sequence are presented in Table 4, and the visualization scene is depicted in Fig. 7c.

|            | book | bowl | hat | cup | chair |
|------------|------|------|-----|-----|-------|
| w/o EIF    | 1.820| 1.211| 2.898| 0.869| 3.086 |
| w/o Reprojection | 0.398| 0.142| 0.443| 0.152| 0.574 |
| Ours       | 0.117| 0.092| 0.224| 0.053| 0.182 |

The metric is the ratio of outliers to inliers
The best results are highlighted in bold

We compare two variants of the proposed method. The first variant solely utilizes the reprojection method for outlier removal. However, due to the continuous motion of the camera, reprojection is unable to handle the outliers introduced by nearby frames, resulting in a considerable decline in performance. The second variant only uses EIF. It is apparent that this geometry-based approach is more robust, which indicates that EIF is crucial for maintaining a point cloud that can approximate the object. The combination of these two methods can further enhance performance.

5.4.2 Rotation Estimation

In contrast to prior work [26] that only relies on the PCA method based on object geometry information, we combine it with a heuristic approach that leverages object appearance. We assess the performance of PCA and our proposed method on yaw estimation for cuboid objects, including books, keyboards and sofas in all sequences. The estimation errors using only the PCA method are 13.3, 11.9 and 14.2, respectively, while the errors are reduced to 5.8, 6.7 and 3.8 after combining the appearance-based line alignment method, resulting in improvements of 56.4%, 43.7% and 73.2%, respectively. Fig. 8 provides an example illustrating the rationale behind

Fig. 8  The line alignment method can effectively estimate the pose of the book. The sparse point cloud is shown in purple. The main direction extracted by PCA method is close to diagonal, which seriously affects the shape estimation of superquadric

(a) book  (b) line alignment  (c) PCA

In contrast to prior work [26] that only relies on the PCA method based on object geometry information, we combine it with a heuristic approach that leverages object appearance. We assess the performance of PCA and our proposed method on yaw estimation for cuboid objects, including books, keyboards and sofas in all sequences. The estimation errors using only the PCA method are 13.3, 11.9 and 14.2, respectively, while the errors are reduced to 5.8, 6.7 and 3.8 after combining the appearance-based line alignment method, resulting in improvements of 56.4%, 43.7% and 73.2%, respectively. Fig. 8 provides an example illustrating the rationale behind
Table 5 Average runtime of system

| Number of objects | Tracking thread (ms) | Mapping thread (ms) |
|-------------------|----------------------|---------------------|
| 0 (ORB-SLAM2)     | 23.7                 | 213.5               |
| 4                 | 27.6 (+3.9)          | 220.9 (+7.4)        |
| 8                 | 30.4 (+6.7)          | 231.6 (+18.1)       |
| 12                | 32.9 (+9.2)          | 236.9 (+23.4)       |
| 16                | 34.6 (+10.9)         | 247.2 (+33.7)       |
| 20                | 37.9 (+14.2)         | 255.6 (+42.1)       |

our design choices. In the case of cuboid objects, the principal directions extracted by PCA often resemble diagonal lines instead of the desired orthogonal edges, leading to imprecise object pose estimation. Moreover, these erroneous poses hinder the accurate fitting of superquadrics to the sparse point clouds, resulting in inaccurate shape estimation. The appearance-based line alignment method proves effective in handling such objects. Therefore, we combine both methods to enhance the robustness of our system.

5.5 Runtime Analysis

Finally, we provide system runtime analysis on a laptop with an Intel Core i5-7400 CPU at 3.0 GHz and 16GB RAM. Notably, the computational time of object detection is not included in the results due to the significant variation in model complexity among different detectors.

The effect of the number of objects on the TUM fr2_desk sequence is analyzed by masking certain object types, with each object being observed an average of 50 times. The results as presented in Table 5, demonstrate that the computational time of both threads positively correlates with the number of objects. Most of the additional time in the tracking thread is dedicated to outlier removal, while the superquadric shape estimation in the mapping thread contributes to the increased computation. In typical scenarios, our algorithm achieves real-time performance at 25-30 HZ. Additionally, our lightweight association strategy has an average processing time of only 0.7 ms per frame.

6 Conclusions

In this work, we propose a novel monocular semantic SLAM system with real-time performance, which utilizes superquadrics with shape parameters to represent objects and construct semantic object maps. We present a separate parameter estimation method capable of adapting to different object shapes and a lightweight data association strategy, both of which leverage the appearance and geometry information of objects. Comprehensive experiments demonstrate the effectiveness and advantages of our method. In the future, we will explore building dense representations of objects using only a monocular camera to further enhance the system’s perception capabilities.

Author Contributions All authors contributed to the study conception and design. Xiao Han collected the data, performed the analysis, and wrote the manuscript. Lu Yang commented on previous versions of the manuscript and critically revised the work. All authors read and approved the final manuscript.

Funding This work was supported by the National Natural Science Foundation of China (Grant No. 61871074).

Code or Data Availability The code will be released upon acceptance. The datasets used are publicly available in:
- TUM RGB-D Dataset: https://vision.in.tum.de/data/datasets/rgbd-dataset
- ICL-NUIM Dataset: http://www.doc.ic.ac.uk/~ahanda/VaFRIC/iclnui m.html
- RGB-D Sences v2 Dataset: https://rgbd-dataset.cs.washington.edu/ dataset/rgbd-scenes-v2/.

Declarations

Conflicts of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Mur-Artal, R., Tardos, J.D.: ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. 33(5), 1255–1262 (2017)
2. Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., Moreno-Noguer, F. (2017) PL-SLAM: Real-time monocular visual SLAM with points and lines. In: 2017 IEEE international conference on robotics and automation, pp. 4503–4508
3. Yunus, R., Li, Y., Tombari, F. (2021) ManhattanSLAM: Robust planar tracking and mapping leveraging mixture of manhattan frames. In: 2021 IEEE international conference on robotics and automation, pp. 6687–6693
4. Martins, R., Bersan, D., Campos, M.F., Nascimento, E.R.: Extending maps with semantic and contextual object information for robot navigation: a learning-based framework using visual and depth cues. J. Intell. Robot. Syst. 99, 555–569 (2020)
5. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016) You Only Look Once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788
6. He, K., Gkioxari, G., Dollar, P., Girshick, R. (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969
7. Qin, Y., Mei, T., Gao, Z., Lin, Z., Song, W., Zhao, X.: RGB-D SLAM in dynamic environments with multilevel semantic mapping. J. Intell. Robot. Syst. 105(4), 90 (2022)
8. Virgolino Soares, J.C., Gattass, M., Meggiolaro, M.A.: CrowdSLAM: visual SLAM towards crowded environments using object detection. J. Intell. Robot. Syst. 102(2), 50 (2021)
9. Wu, Y., Zhang, Y., Zhu, D., Chen, X., Coleman, S., Sun, W., Hu, X., Deng, Z. (2021) Object SLAM-based active mapping and robotic navigation. J. Intell. Robot. Syst. 105(4), 90 (2022)
grasping. In: 2021 international conference on 3D vision, pp. 1372–1381

10. Qian, Z., Fu, J., Xiao, J.: Towards accurate loop closure detection in semantic SLAM with 3D semantic covisibility graphs. IEEE Robot. Auton. Lett. 7(2), 2455–2462 (2022)

11. Zins, M., Simon, G., Berger, M.-O. (2022) OA-SLAM: Leveraging objects for camera relocalization in visual SLAM. In: 2022 IEEE international symposium on mixed and augmented reality, pp. 720–728

12. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J. (2013) SLAM++: Simultaneous localisation and mapping at the level of objects. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1352–1359

13. Yang, S., Scherer, S.: CubeSLAM: Monocular 3-D object SLAM. IEEE Trans. Robot. 35(4), 925–938 (2019)

14. Wu, Y., Zhang, Y., Zhu, D., Feng, Y., Coleman, S., Kerr, D. (2020) EAO-SLAM: Monocular semi-dense object SLAM based on ensemble data association. In: 2020 IEEE/RSJ international conference on intelligent robots and systems, pp. 4966–4973

15. Hosseinzadeh, M., Li, K., Latif, Y., Reid, I. (2019) Real-time monocular object-model aware sparse SLAM. In: 2019 IEEE international conference on robotics and automation, pp. 7123–7129

16. Ok, K., Liu, K., Frey, K., How, J.P., Roy, N. (2019) Robust object-based SLAM for high-speed autonomous navigation. In: 2019 IEEE international conference on robotics and automation, pp. 669–675

17. Nicholson, L., Milford, M., Sünserhauf, N.; QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM. IEEE Robot. Autom. Lett. 4(1), 1–8 (2018)

18. Runz, M., Blatter, M., Agapito, L. (2018) MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In: 2018 IEEE international symposium on mixed and augmented reality, pp. 10–20

19. Sünserhauf, N., Pham, T.T., Latif, Y., Milford, M., Reid, I. (2017) Meaningful maps with object-oriented semantic mapping. In: 2017 IEEE/RSJ international conference on intelligent robots and systems, pp. 5079–5085

20. Rubino, C., Crocco, M., Del Bue, A.: 3D object localisation from multi-view image detections. IEEE Trans. Pattern Anal. Mach. Intell. 40(6), 1281–1294 (2017)

21. Tian, R., Zhang, Y., Feng, Y., Yang, L., Cao, Z., Coleman, S., Kerr, D.: Accurate and robust object SLAM with 3D quadric landmark reconstruction in outdoors. IEEE Robot. Autom. Lett. 7(2), 1534–1541 (2022)

22. Liao, Z., Hu, Y., Zhang, J., Qi, X., Zhang, X., Wang, W.: SO-SLAM: Semantic object SLAM with scale proportional and symmetrical texture constraints. IEEE Robot. Auton. Lett. 7(2), 4008–4015 (2022)

23. Hu, Y., Wang, W.: Making parameterization and constraints of object landmark globally consistent via SPD (3) manifold. IEEE Robot. Autom. Lett. 7(3), 6383–6390 (2022)

24. Rosinol, A., Violante, A., Aabte, M., Hughes, N., Chang, Y., Shi, J., Gupta, A., Carlone, L.: Kiema: From SLAM to spatial perception with 3D dynamic scene graphs. Int. J. Robot. Res. 40(12–14), 1510–1546 (2021)

25. Zhen, W., Yu, H., Hu, Y., Scherer, S. (2022) Unified representation of geometric primitives for Graph-SLAM optimization using decomposed quadrics. In: 2022 IEEE international conference on robotics and automation, pp. 5636–5642

26. Tschopp, F., Nieto, J., Siegwart, R., Cadena, C. (2021) Superquadric object representation for optimization-based semantic SLAM. arXiv:2109.0962

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.
Xiao Han received the bachelor’s degree in automation from Chongqing University, Chongqing, China, in 2021. He is currently pursuing the master’s degree in control science and engineering with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include computer vision, SLAM, and scene semantic understanding.

Lu Yang received the bachelor’s degree from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2005 and the Ph.D. degree in electrical engineering and computer science from Nagoya University, Nagoya, Japan, in 2011. He joined the School of Automation Engineering, UESTC, in 2011, as a Lecturer. Since 2022, he has been a Professor with UESTC. His research interests include computer vision, pattern recognition, machine learning, and 3-D multimedia applications. Prof. Yang was the Local Arrangement Chair of the IEEE International Conference on Multimedia & Expo in 2014.