applied sciences

MDPI

Article

Detect Orientation of Symmetric Objects from Monocular Camera to Enhance Landmark Estimations in Object SLAM

Zehua Fang [1], Jinglin Han [2] and Wei Wang [3,*]

[1] School of Mathematics and Science, Beihang University, Beijing 100191, China
[2] School of Integrated Circuit Science and Engineering, Beihang University, Beijing 100191, China
[3] School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China
* Correspondence: wangweilab@buaa.edu.cn

Abstract: Object simultaneous localization and mapping (SLAM) introduces object-level landmarks to the map and helps robots to further perceive their surroundings. As one of the most preferred landmark representations, ellipsoid has a dense mathematical expression and can represent the occupied space of objects with high accuracy. However, the orientations of ellipsoid approximations often fail to coincide with the orientation of objects. To further improve the performance of object SLAM systems with ellipsoid landmarks, we innovatively propose a strategy that first extracts the orientations of those symmetric human-made objects in a single frame and then implements the results of the orientation as a back-end constraint factor of the ellipsoid landmarks. Experimental results obtained show that, compared with the baseline, the proposed orientation detection method can reduce the orientation error by more than 46.5% in most tested datasets and improves the accuracy of mapping. The average translation, rotation and shape error improved by 63.4%, 61.7% and 42.4%, respectively, compared with quadric-SLAM. With only 9 ms additional time cost of each frame, the object SLAM system integrated with our proposed method can still run in real time.

Keywords: object SLAM; ellipsoid landmarks; orientation detection

Citation: Fang, Z.; Han, J.; Wang, W. Detect Orientation of Symmetric Objects from Monocular Camera to Enhance Landmark Estimations in Object SLAM. *Appl. Sci.* **2023**, *13*, 2096. https://doi.org/10.3390/app13042096

Academic Editor: Nuno Silva

Received: 11 January 2023
Revised: 3 February 2023
Accepted: 4 February 2023
Published: 6 February 2023

1. Introduction

Robotics provides extensive capabilities in the fields of manufacturing, service, communications, and allied areas [1]. As one of the most active research fields in robotics, simultaneous localization and mapping (SLAM) has enabled robots to localize themselves in most unknown environments based on LiDAR [2,3] or vision [4,5] after development over two decades. However, in these SLAM systems, robots initialize and optimize low-level landmarks, such as points and grids, mostly for self-localization instead of truly perceiving, learning and interacting with their surroundings.

Recently, deep-learning-based image detectors have allowed robots to recognize objects which can be utilized as semantic landmarks and significantly improved robots' performances of perception [6]. The methods to integrate SLAM with object-level semantic information can be classified into two main types. One is to add semantic labels to the dense or semi-dense maps after they are constructed by other systems [7,8]; the objects can be represented by a cluster of points with the same semantic labels. Another type, in contrast, does not need any existing maps. This type directly infers 3D landmarks from 2D image-detection results using prediction models, similar to a differently designed end-to-end neural network [9], or geometrical reasoning based on multi-view observation, such as cuboids [10] and ellipsoids [11], to obtain the abstract shapes and represent the geometry of objects. These methods are more lightweight and flexible because only a coarse trajectory of the camera is needed.

Ellipsoids, due to their property of being compactly parameterized and easily manipulated within the framework of projective geometry, have been chosen by more and more

researchers as approximations of object landmarks. By adding geometrical constraints and priors to the ellipsoids, the accuracy and robustness of mapping can be further improved [12,13]. The curved surfaces of ellipsoids enable them to easily fit most 3D objects, but also make them less sensitive to the orientation of the objects. Specifically, an ellipsoid can still nicely wrap around an object even if its principal axes are not parallel to the orientation of the object. Consequently, the orientations of most ellipsoid landmarks are not meaningful in the real world. However, in practical applications, the orientations of object landmarks can be very informative—outdoors they can help infer the direction of motion of an object, while indoors they can help the robot further understand object-human interactions. For example, a robot with object orientation information can navigate to a certain direction of a chair if needed, or keep away from the front side of a TV when someone is watching it.

The author of [14] proposed an orientation factor for quadric-SLAM, in which only the horizontal or the vertical principal orientation of the objects was integrated. However, there is one more degree of freedom that needs to be further constrained with additional information, i.e., to determine in what direction a horizontally placed object is facing. We noticed that most human-made objects, such as chairs, sofas, keyboards, etc., are built to be symmetrical for convenience, so we consider their symmetry planes to be an effective constraint for the orientations of landmark objects. Liao's previous work [15] was the first attempt in mapping with the assistance of an object symmetry feature. The method in [15] requires the landmarks to be extracted in advance—then their symmetry axes are examined with the axes of the corresponding ellipsoid approximations. This framework is actually a verificationinstead of a measurement of the object orientation. Thus, all images from the front end must be preserved and iteratively read in our back-end optimization, causing huge time and memory consumption.

In this work, we obtain the symmetry planes of the objects directly from the single-frame monocular RGB image and integrate the results in the back-end optimization as constraints. The greatest challenge of this method is that a symmetrical 3D object may not be symmetrical in the 2D image when the camera does not face the front of the object. To address this challenge, we herein propose an original strategy. The contributions of this work are summarized as follows:

- We propose a projection restoration method to estimate the 3D symmetry plane of an object from the 2D image.
- We integrate object symmetry planes in the factor graph of object SLAM systems to improve the orientation accuracy of ellipsoid landmarks.
- Based on the above two points, we propose a lightweight real-time object-level mapping system.

In Section 2, we discuss prior work in object landmark representation and the implementation of symmetry features in SLAM. The method to estimate the 3D symmetry plane of the objects and the framework of the overall SLAM system is described in Section 3. In Section 4, we demonstrate the accuracy of the proposed single-frame orientation extraction method and the mapping system with a variety of experimental results. Finally, the conclusions are drawn in Section 5.

2. Related Work

2.1. SLAM with Object-Level Landmarks

Compared with a traditional visual-SLAM framework that provides maps containing points or line features, object SLAM integrates in-frame object detection into measurement, resulting in more meaningful mapping and more robust localization results [8]. In recent years, the object-detection model based on deep learning has shown potential in SLAM. Arunabha et al. [16] proposed the WilDect–YOLO detection model, which can realize automatic high-performance object detection. Aisha et al. [17] proposed a precise single-stage detector (PSSD) by adding extra layers to single-stage object detectors (SSD). The multiple graph learning neural networks (MGLNN) proposed by Jiang et al. achieved

multiple graph-learning and multi-view semi-supervised classification [18]. In studies of object SLAM, various representations have been used to model the objects in the scene. The authors of [19] proposed an object-SLAM system using an RGB-D camera by matching CAD models of tables and chairs in the database. In [7], proposed by Martin et al., every point in the dense map carries a semantic label given by the pixel segmentation to make the robot object-aware. Yang et al. [10] proposed cube-SLAM that described objects with a cuboid bounding box, while ellipsoids were used in [11] to approximate the position and size of objects. The above studies show that the geometrical model used in object-SLAM is evolving to be more lightweight, computation-friendly and compatible for general objects.

The low computational complexity is obtained by a reduction in fineness of the object model. To overcome the loss of model accuracy that limits the performance of dual quadric-based object SLAM, a variety of geometric properties have been implemented as prior constraints to improve the robustness and accuracy of the systems. The authors of [12] proposed that objects should stand on a plane instead of floating in the air. In [13], the authors introduced a texture plane to avoid the observability problem during common forward-translating camera movements, while Chen et al. [20] proposed a different dual quadric initialization method to solve similar issues. Furthermore, attempts have also been made to extend the range of object representations, such as introducing super-ellipsoids to unify cuboids and ellipsoids [21] and using a pre-trained variational auto encoder as an efficient and optimizable object descriptor [22]. Meanwhile, the quadric-based method still combines well with these new methods and can act as an initial approximation, providing a coarse-to-fine estimation of objects [23].

2.2. The Symmetry of Objects

Symmetry is a common property in human-made objects and has been applied to refine the point cloud results of 3D reconstruction [24,25]. Thus, it shows great potential to further improve monocular object-level SLAM. The detection of symmetry patterns in 2D images has been thoroughly studied in recent years. Both the CVPR conference and ICCV workshops have organized competitions on single and multiple symmetry axes detection in natural images [26,27]. The authors of [28] obtained the symmetry axes with constellations of interested points detected with a rotation invariant feature and achieved the best results in the 2013 competition. Marcelo et al. [29] outperformed other competitors in the 2017 competition with a registration technique that registered the collection of original points to their mirror-reflected counterparts. Methods based on image gradient [30] or important edges also achieved robust symmetry axes extraction [31–33].

It is worth noting that the aforementioned works focus on detecting the existing symmetry in the images whose pixel planes are nearly parallel to the symmetrical "front" sides of the objects. However, as Figure 1c shows, from the perspective of a monocular camera in a SLAM system, this is not guaranteed, since there is often an angle—which is exactly what we aim to find in this work—between the object orientation and the camera axis. This difference makes the objects look unsymmetrical or even distorted in the 2D frames that we process.

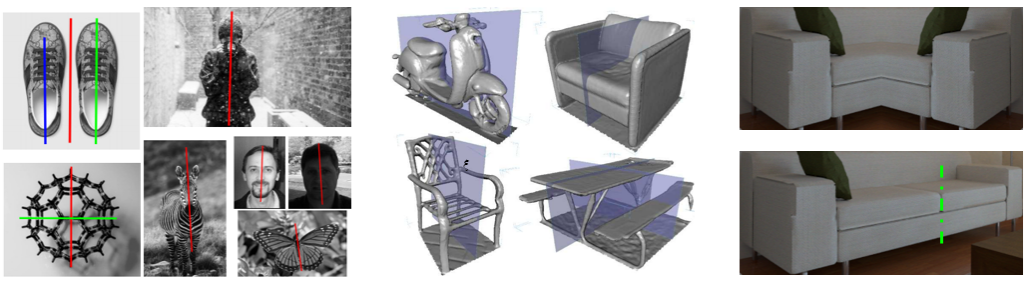

Figure 1. (**a**) Symmetry in 2D image [34]. (**b**) Symmetry in 3D world [27]. (**c**) 3D-symmetrical object may not be symmetrical in 2D image

3. Materials and Methods

Artificial objects are often made with symmetry, such as chairs, cars, and keyboards. However, as we addressed above, the 3D symmetry no longer holds on the 2D pixel plane when the camera is not fronto-parallel along the object. In this Section, we propose a method to estimate the main direction of these objects by the distorted 2D symmetry even when the camera is not exactly facing the front direction of the object.

3.1. Description of Ideal Pixel Symmetry

We start with the simplest case: the object is symmetric in the pixel plane. This situation occurs when the camera is directly on the front side of the object. Suppose the symmetry axis of the object on the image is $l_s = [1, 0, -c_x]^T$, then, for any pixel point $p = [p_x, p_y, 1]^T$ on the image, its symmetry point is $p' = [2c_x - p_x, p_y, 1]^T$. Let

$$\mathbb{S} = \begin{bmatrix} -1 & 0 & 2c_x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

we have $p' = \mathbb{S}p$. Ideally, as Figure 2 shows, the pixels in the object detection box of this frame should be strictly symmetric: the textures (e.g., gray-scale values, RGB values) at p and p' should be equal.

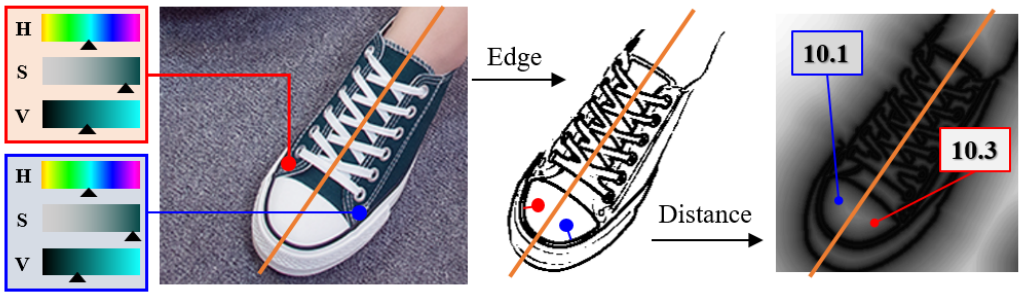

Figure 2. On an ideal 2D symmetric image, the hue, saturation, gray-scale value and the distance to the nearest edge of two symmetric pixels should be equal.

For robustness and speed in practical use, we choose the HSV (hue, saturation, value) color space to describe the pixel p itself, noted as $\mathrm{H}(p), \mathrm{S}(p), \mathrm{V}(p)$. We use the DT (distance transform) values as a kind of texture descriptor near the pixel p, noted as $\mathrm{DT}(p)$. The combined descriptors are constructed as:

$$Sym(p) = \alpha \mathrm{H}(p) + \beta \mathrm{S}(p) + \delta \mathrm{V}(p) + \eta \mathrm{DT}(p) \tag{2}$$

where $\alpha = \beta = 0.5\delta = 2\eta$. So, in the ideal front-view case, we should have:

$$f_{sym} = \sum_i \| Sym(p_i) - Sym(p_i') \| = 0 \tag{3}$$

3.2. Projection to Recover Symmetry

Since the camera's optical axis direction n_C is often not directly along the object in real situations, we need to transform the pixels inside the object bounding box by an approximate projective transformation. Thus, the problem can be converted to an ideal symmetric description problem, which has been discussed in Section 3.1. Specifically, we assume that the object is placed on the ground, so that the positive direction of the object n_O must be parallel to the horizontal, which can then be expressed in terms of the angle θ between the x-axis of the object's coordinate system and the x-axis of a fixed world coordinate system. Next, we try to simulate the observation of the camera rotating around its center, vertically to the z-axis of the world coordinate system in $[-45, 45]$ degree, and

evaluate the symmetry of the object in the 2D image plane at that simulated viewpoint to determine the direction of the symmetry plane. When the recovered image best satisfies the symmetry in two dimensions, the simulated n_C should, theoretically, be most aligned with n_O at this point.

There are two reasons why a pure rotation of the camera around its center is chosen. One is that the three-dimensional position of the object in the world is unknown at the time of obtaining the camera image, so only the rotational distortion can be corrected. The other is that the transformation between pixels on the image before and after the conjugate rotations R_c^n has a simple homographic form, i.e.,

$$v^n = (KR_c^n K^{-1})v^c \tag{4}$$

where K denotes the camera intrinsics matrix, which describes the relationship between object points and image points.

It is worth noting that the camera does not rotate around its own z-axis, but around the z-axis of the world coordinate system. Let $Rot(Z)$ be the rotation matrix corresponding to the rotation θ of any coordinate system along its own z-axis, then the rotation transformation matrix of the camera is an adjoint transformationwith the form $R_c^n = R_w^c \cdot RotZ(\theta) \cdot (R_w^c)^{-1}$. Let

$$\mathbb{P}(\theta) = KR_w^c \cdot RotZ(\theta) \cdot (R_w^c)^{-1}K^{-1} \tag{5}$$

then we have $v^n = \mathbb{P}v^c$. If the camera is rotated to face exactly the front side of the object, the corrected image should satisfy the 2D direct symmetry, as shown in Figure 3.

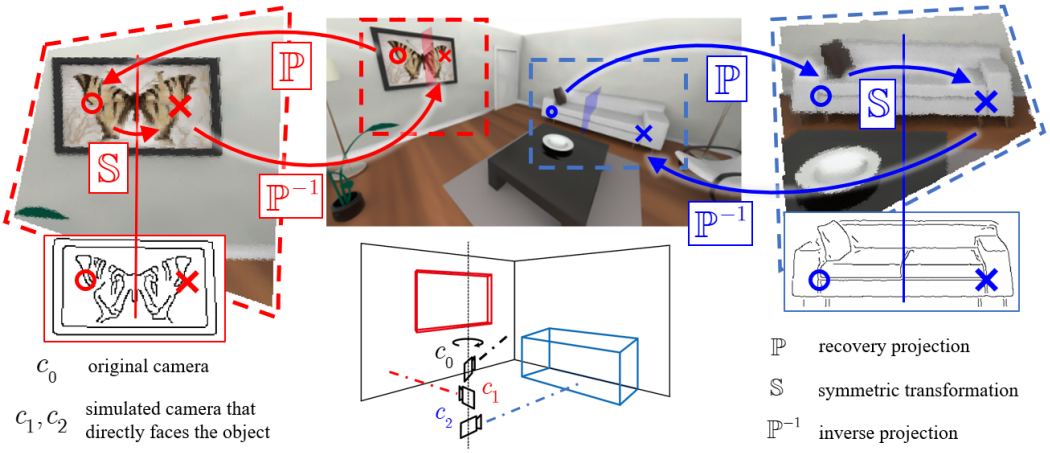

Figure 3. A demonstration of our algorithm. In the room, both the sofa and the wall painting are symmetrical in the 3D world. However, they are not symmetrical in the 2D image because the camera does not directly look at them. We simulate the case when the camera is directly facing their front side by applying two projective transformations to each image, respectively, and then we evaluate the 2D symmetry.

We show the source image projectively warped by homography $\mathbb{P}(\theta)$ only to better illustrate the result of the original image after the projective transformation. In practice, since we need to traverse all θ, our comparison is actually performed on the original image (without actually computing the projectively warped result of the original image). The method to obtain the corresponding symmetry point on the original image is shown in the middle of Figure 3, i.e., for a point p^c, first obtain the position of the image point $p^n = \mathbb{P}p^c$ corresponding to the simulated rotation, then obtain its symmetry point $p_{sym}^n = \mathbb{S} \cdot p^n$, and,

finally, back-project the point to the original image $p^c_{sym} = \mathbb{P}^{-1} p^n_{sym} = (\mathbb{P}^{-1}\mathbb{S}\mathbb{P})p^c$. So, we are actually solving

$$\underset{\theta\in(-45°,45°)}{\arg\min} \sum_i ||Sym(p_i) - Sym((\mathbb{P}^{-1}\mathbb{S}\mathbb{P})p_i)|| \tag{6}$$

3.3. Obtain the Sampling Point and Symmetry Axis

The sampling points are selected as shown in Figure 4. Since the DT transformation value is the distance between the point and the nearest edge point to itself, most of the sampled points are selected on the canny edges and a few points are selected randomly to increase the robustness. Under this strategy, the minimum function we solve has a clear meaning—since symmetric objects have symmetric edges, we want the symmetric points at the edge points of the image to remain on the edge, or at least as close to the edge as possible.

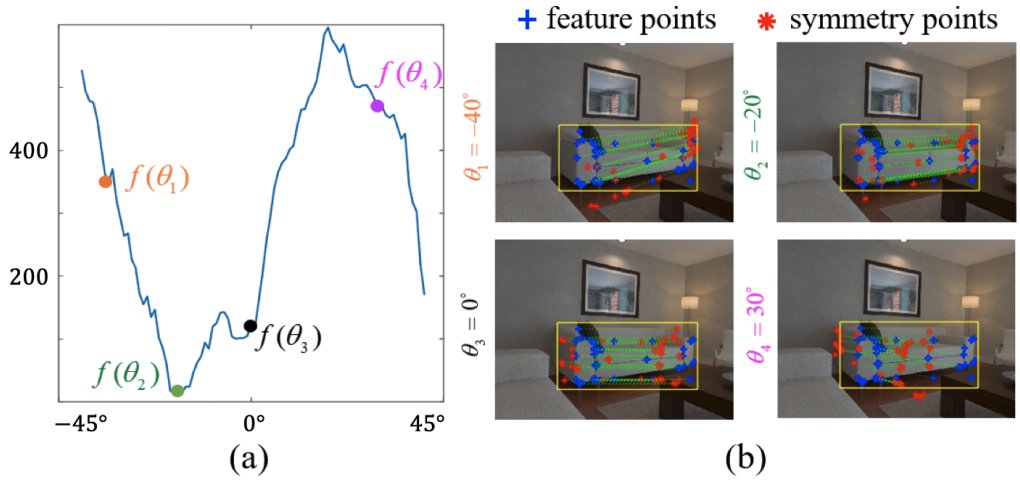

Figure 4. Estimation of orientation by minimizing the cost function. (**a**) The cost function value with respect to orientation angle. (**b**) The corresponding symmetry points under different angles. The top right is where the function reaches the minimum.

We can see that the object detection box of the original image is no longer a rectangle after the projective transformation $\mathbb{P}$, but we can still get the value of c_x in $\mathbb{S}$ by the center of the four corner points of the detection box $\{b_i\}$ after the transformation.

$$\begin{bmatrix} c_x \\ c_y \\ 1 \end{bmatrix} = \sum_{i=1}^{4} \frac{\mathbb{P} \cdot b_i}{4} \tag{7}$$

We sample the rotation angles θ at 5-degree intervals, calculate the value of the cost function $\{f_i(\theta)|i = 1, 2 \cdots 18\}$ in Equation (6) and note the angle that minimizes the cost function as the object orientation. In addition, the observation is considered valid only if the minimum value is less than $0.1 \sum_i f_i/18$.

3.4. System Overview

We solve the object SLAM problem for all ellipsoidal approximations of objects $\mathcal{Q} = \{Q^*_j\}^J_{j=0}$ with J object landmarks and T poses of the camera $\mathcal{X} = \{x_t | x_t \in \mathrm{SE}(3)\}^T_{t=0}$. The front-end input of the system includes the monocular images, as well as odometry data from an external vision-based localization system. The object detector extracts bounding boxes from the RGB images. We extract K bounding box measurements of objects $\mathcal{B} = \{B_k | B_k = [x_{min}, y_{min}, x_{max}, y_{max}]\}^K_{k=0}$ along with the semantic class labels $\mathcal{C} = \{c_k\}^K_{k=0}$.

The following data association method solves the correspondence between bounding boxes $\mathcal{B}$ and landmarks $\mathcal{Q}$. Although more sophisticated data association [35,36] can be used instead, in this work, we use a minimal technique: each valid bounding box detection is associated to an existing ellipsoid or triggers a new landmark creation. Every landmark approximation ellipsoid is first back-projected into the frame as a rectangular landmark box using the measurement model proposed in [11]. Given a new bounding box detection B_k, we synthesize a cost value for matching it with the existing object landmark Q_j, which is defined as:

$$cost\left(B_k, Q_j\right) = \lambda * d_{k,j} + \mu * IoU_{k,j} + \xi * l_{k,j} \tag{8}$$

where $d_{k,j}$ is the distance between the centroids of the bounding box and the back-projected landmark box, $IoU_{k,j}$ is the intersection over union (IoU) of the two boxes, and $l_{k,j}$ measures the semantic discrepancy between the detection and the landmark. $l_{k,j}$ is set to 0 if the detection and the landmark have the same semantic label, and 1 if their labels are different. Here, we let λ be the reciprocal of the image diagonal length and $\mu = \xi$. Then, with the Hungarian matching algorithm [37], we associate the detection with the landmark when the cost reaches the lowest, and, if no match is found, we initialize a new landmark.

With the data association problem solved, the problem in the back end of the object SLAM can be written as:

$$\begin{aligned} P(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Q}} \mid \boldsymbol{\mathcal{B}}, \boldsymbol{\Theta}^s, \boldsymbol{\mathcal{I}}, \mathcal{C}) \propto & \underbrace{\prod_{k=0}^{K} P\left(\boldsymbol{B}_k \mid \boldsymbol{Q}_{j_k}^*, \boldsymbol{x}_{t_k}\right)}_{\text{Bounding Box [11]}} \\ & \underbrace{\prod_{d=0}^{D} P\left(\boldsymbol{\theta}_s^t \mid \boldsymbol{Q}_{j_d}^*, \boldsymbol{x}_{t_d}\right)}_{\text{Symmetry (Ours)}} \underbrace{\prod_{j=0}^{J} P\left(\boldsymbol{Q}_j^* \mid c_j\right)}_{\text{Semantic Prior [13]}} \underbrace{\prod_{t=0}^{T} P(\boldsymbol{x}_t \mid \boldsymbol{I}_{0:t})}_{\text{Pose Prior}}. \end{aligned} \tag{9}$$

The problem can also be formed as a factor graph, including nodes composed of objects and camera poses and edges composed of observation constraints, as in Figure 5, where X is the camera poses and Q is the objects in the map. z is the camera-object observation constraint, o is the odometry constraint; both were described in detail in [11]. s is the newly added 3D symmetric constraints and will be emphasized in the next subsection.

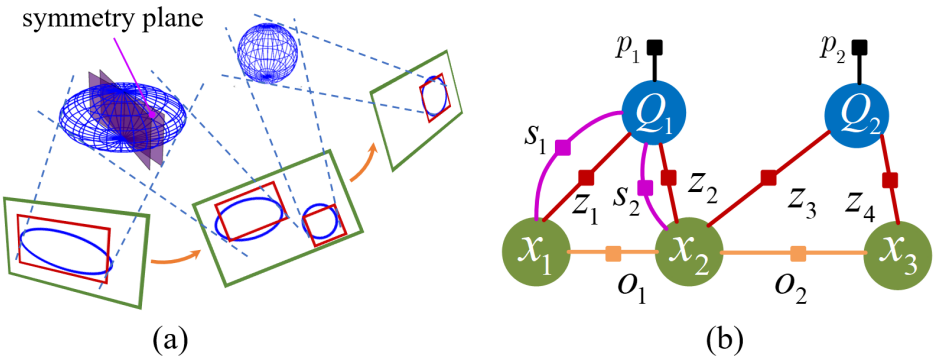

Figure 5. (**a**) A simplified demonstration of the object-SLAM process containing two objects and three frames, where the objects on the left are symmetrical. (**b**) The corresponding factor graph, where our added symmetric factors are marked in purple.

Finally, we can obtain the optimal estimation of camera poses $\hat{\mathcal{X}}$ and objects $\hat{\mathcal{Q}}$ by maximizing the posterior probability.

$$\hat{\mathcal{X}}, \hat{\mathcal{Q}} = \arg\max_{\mathcal{X}, \mathcal{Q}} P(\mathcal{X}, \mathcal{Q} \mid \mathcal{B}, \boldsymbol{\Theta}^s, \mathcal{I}, \mathcal{C}) \tag{10}$$

3.5. Factor Formulation

Assuming Gaussian measurement and process models, we can write (9) as a nonlinear least-squares problem:

$$\begin{aligned}\hat{\mathcal{X}}, \hat{\mathcal{Q}} &= \arg\min_{\mathcal{X},\mathcal{Q}} -\log P(\mathcal{X}, \mathcal{Q} \mid \mathcal{B}, \Theta^s, \mathcal{I}, \mathcal{C}) \\ &= \arg\min_{\mathcal{X},\mathcal{Q}} \Bigg\{ \sum_{t=0}^{T} \| E_{Odom}(\bar{\mathbf{x}}_t, \mathbf{x}_t) \|^2_{\Sigma_o} + \\ &\sum_{k=0}^{K} \left\| E_{Box}\left(\mathbf{Q}^*_{j_k}, \mathbf{x}_{t_k}, \bar{\mathbf{B}}_k\right) \right\|^2_{\Sigma_b} + \\ &\sum_{d=0}^{D} \left\| E_{Label}\left(\mathbf{Q}^*_{j_d}, c_j\right) \right\|^2_{\Sigma_l} + \\ &\sum_{j=0}^{J} \left\| E_{Sym}\left(\mathbf{Q}^*_{j_d}, \theta_j\right) \right\|^2_{\Sigma_s} \Bigg\},\end{aligned} \tag{11}$$

where $E_{Odom}, E_{Box}, B_{Label}$ are odometry factors, bounding box measurement factors and semantic prior factors, respectively. The formulation of these factors has been discussed comprehensively in [13]. E_{Sym} is the new symmetric factor and should reflect the error between the symmetry plane of the estimated landmark Q and its detected symmetry planes $\{\theta_j\}$. However, because of the ambiguity of the object coordinate system, as shown in Figure 6a, the error cannot be simply defined as the subtraction of two angles. Instead, it should relate to the minimum rotation angle required to align the estimated object's x-ory-axes parallel to the measured symmetry plane. We define the error function of the symmetric factor as

$$E_{Sym} = \sin 2(\theta_O - \theta_S) \tag{12}$$

With a period of $\pi/2$, as shown in Figure 6b, the value of the function is 0 when θ_o is in the same or opposite direction as the x- or y-axes, and takes the maximum value when it is not close to either of the two axes, satisfying the above requirements while having good derivability.

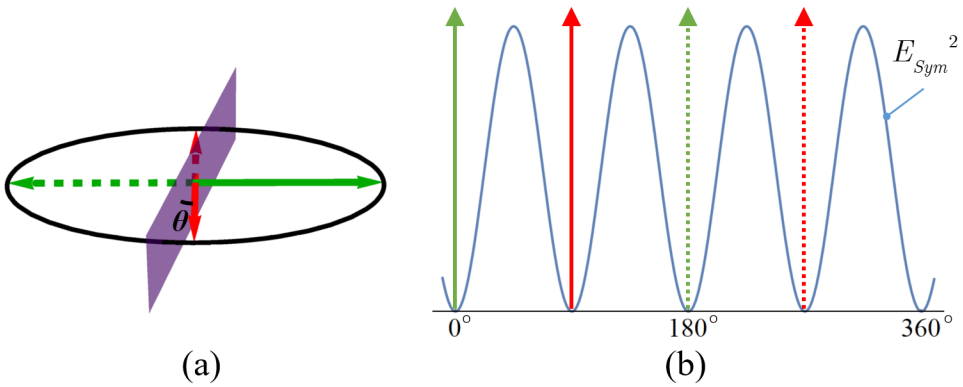

Figure 6. (**a**) The error of the ellipsoid and its observed symmetry plane should be related to the minimum rotation angle required to align the two of them. (**b**) The squared value of the symmetry error function E_{Sym}.

4. Results

To fully verify the single-frame orientation estimation and the complete mapping performance with the orientation factor proposed in this paper, we conducted experiments on both public datasets and author-recorded real robot datasets, the TUM RGB-D [38] and ICL-NUIM [39] datasets, which are widely used in SLAM, covering both room-level

and desktop-level environments. To better reflect the effectiveness on the mobile robot, we conducted experiments on a turtlebot3 with a Kinect camera operating in a home-like environment. Although our method only needed RGB channels in the experiments, we used the result of ORB-SLAM2 [4] with depth information as odometry data to avoid scale drift. We used yolo-v3 [40] as the object detector.

4.1. Single-Frame Symmetry Plane Estimation

To evaluate the effectiveness of our proposed method for estimating the orientation of symmetric objects based on single-frame images, we took each valid observation of the symmetric object orientation before multi-frame optimization and computed their average orientation errors with respect to the ground-truth value, i.e, the minimum rotation angle required to align the estimated object's three rotation axes with any axis of the ground-truth object to a straight line.

Two methods were chosen as our benchmark. One was quadric-SLAM [11], which only considers the position and size of the object detection box, without considering the specific texture in the detection box. Therefore, although the SVD (singular value decomposition) method can be used to obtain an initialization estimate of the ellipsoid, the orientation of the ellipsoid principal axis is not meaningful in practice. The other method is the object initialization method used in cube-SLAM [10], which considers the line features of the object image and, in turn, infers the orientation of the object.

We need to point out that these methods are not strictly similar—quadric-SLAM requires at least three frames to initialize the object, and cube-SLAM obtains not only the orientation, but also the position and size of the object by single-frame inference. So we actually give quadric-SLAM more than one frame of data and record the results of quadric-SLAM initialization and the results of multi-frame optimization. Moreover, since this subsection discusses the effectiveness of our single-frame orientation observation method, we only compare the object orientation accuracy—the overall accuracy of the object landmarks will be discussed in the next subsection.

The measurement results of our method and [10] using the same images are shown in Figure 7. Since the orientation measurement method of [10] relies mainly on the vanish points of straight-edged lines, its results will be inaccurate when there is too much texture inside the object (TUM-keyboard) or when the edges of the object are curved (ICL-chair). On the contrary, our method can still obtain the exact orientation in the above cases, because the symmetry property still exists. However, we also found some failure cases, such as the third column of Figure 7, where the chair was mistaken for a diamond shape by our method and, in turn, received an incorrect estimated orientation, due to the ambiguous information of the single frame observation.

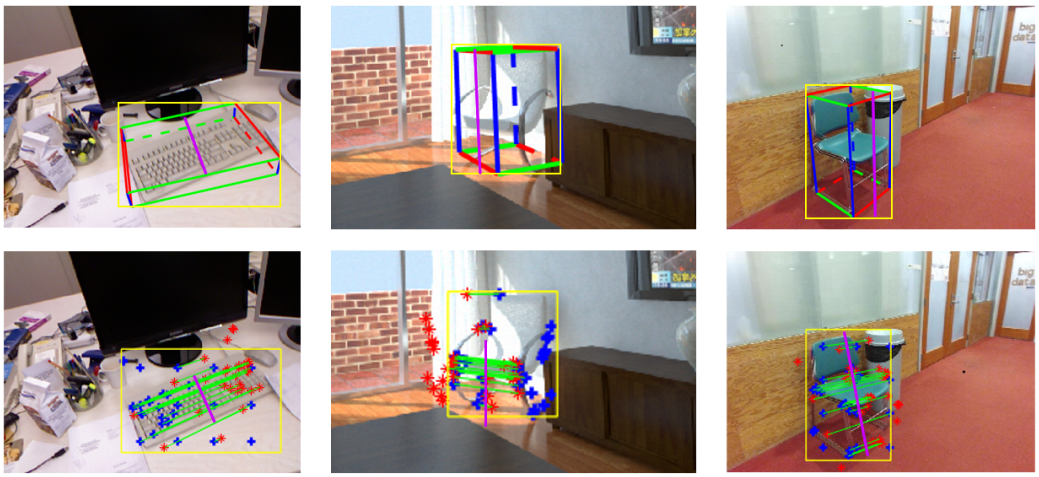

Figure 7. Single image 3D orientation detection examples. The first row is the result using [10]. The second row is our result.

The quantitative results are presented in Table 1. As quadric-SLAM does not explicitly constrain the orientation of the object, the average orientation error is relatively large, reaching 38.7 degrees and 33.0 degrees, respectively. Cube-SLAM achieves the most accurate orientation in the case of ideal cube-like objects, such as cabinets and books, and we outperform other methods in most of the remaining cases. The orientation error of our method is reduced by 60.6% compared to quadric-SLAM and 46.5% compared to cube-SLAM, reaching an average error of 13.07 degrees.

Table 1. Error of single frame orientation detection (°).

Datasets	Label	Quadric-SVD	Quadric-SLAM	Cube-SLAM	Ours
ICL_home	Bench	38.14	31.91	8.20	**5.82**
	Chair1	38.08	41.65	17.17	**3.73**
	Chair2	34.79	43.22	35.78	**5.45**
Fr1_desk	TV	49.30	49.30	33.96	**17.51**
	Book	38.46	40.09	**14.36**	19.75
	Keyboard	34.70	36.50	19.74	**6.97**
Fr2_desk	Mouse	54.11	59.13	**22.03**	29.08
	Keyboard	40.82	24.89	15.31	**10.72**
	TV	41.65	20.88	42.76	**19.22**
	Book	53.37	44.96	**7.28**	12.85
Fr3_cabi	Cabinet	19.98	18.31	**4.17**	11.44
Real_robot	Chair	10.87	11.31	24.73	**10.07**
	Bench1	41.53	**11.01**	36.64	11.58
	Bed	52.38	30.67	43.18	**24.62**
	TV	31.92	30.58	40.09	**7.28**
Average		38.67	32.96	24.36	**13.07**

In summary, the experiments demonstrate that the properties of symmetry can help achieve finer orientation estimation of the object landmark. Considering that the vanishing point-based estimation method is more effective on small and square objects, combining the two methods may lead to more accurate results

4.2. Multi-Frame Object-Mapping

Indoors, the visual odometry (or SLAM) system has been able to give fairly accurate trajectory estimates. Since we are mainly concerned with the accuracy of object landmark establishment, we fix all the camera nodes in the back-end optimization and optimize only the 9-DOF object landmark. The mapping results are shown in Figure 8.

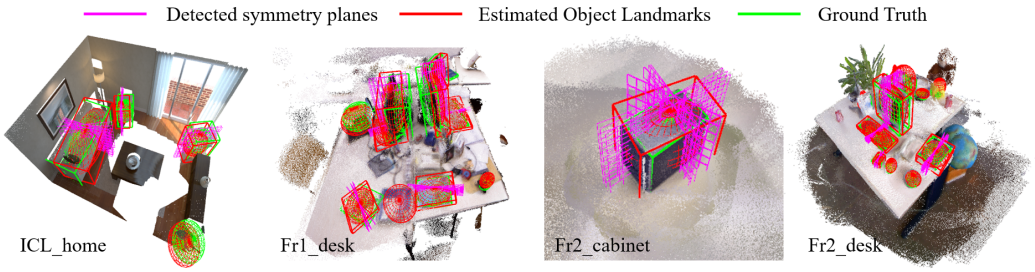

Figure 8. Landmark estimation results from different datasets, the ground-truth objects and point clouds are shown as references. For symmetric objects, the circumscribed cubes of ellipsoidal landmarks are shown to better demonstrate orientation. A part of the detected symmetry planes is marked in purple.

We use the indicators translation, rotation and shape error to fully evaluate the mapping effects. The translation error (m) measures the center distance between the estimated

object and the ground-truth object. The shape error evaluates the 3D intersection over union (IoU) between the two circumscribed cubes after aligning the center and rotation of the estimated object and the ground-truth object. For objects with symmetry, rotation error (deg) is the orientation error defined in the subsection above.

We conducted experiments using the original SVD initialization and joint optimization method in quadric-SLAM, as well as our method with the symmetry factor. The shape error of cube-SLAM is cited from [10]. As shown in Table 2, the translation and shape of quadric-SLAM after joint optimization is improved compared to SVD initialization. With the symmetry factor added in the back-end factor graph, our method achieved better average translation, rotation and shape error values of 0.15 m/12.93 deg/0.47, improved by 63.4%/61.7%/42.4%, respectively, compared with quadric-SLAM. In the Fr3_cabinet, the result of cube-SLAM obtained an IoU of 0.64, 48% higher than ours, as the cabinet has a tight cubic shape without other texture disturbance. For the results with the other tested datasets, we outperformed the two SOTA object-level SLAM systems.

Table 2. Object-mapping results. The translation, rotation and shape error are calculated and denoted as T, R, S.

Datasets	Quadric-SVD			Quadric-SLAM			Cube-SLAM	Ours		
	T	R	S	T	R	S	S	T	R	S
ICL_home	0.85	39.15	0.10	0.29	37.14	0.30	0.49	**0.21**	**6.14**	**0.65**
Fr1_desk	0.40	46.99	0.31	0.33	47.39	0.40	-	**0.08**	**12.21**	**0.45**
Fr2_desk	0.34	36.70	0.25	0.24	38.13	0.42	-	**0.10**	**11.85**	**0.48**
Fr3_cabinet	0.06	19.98	0.34	0.05	18.31	0.33	**0.64**	**0.05**	**9.87**	0.43
Real_robot	1.58	29.85	0.24	1.14	25.51	0.19	-	**0.31**	**24.60**	**0.35**
Average	0.65	34.53	0.25	0.41	33.30	0.33	-	**0.15**	**12.93**	**0.47**

We implemented the algorithm in C++ and used the gtsam library for the graph optimization. The object detection runs on GTX 1660s with 33 Hz and can work in parallel with other visual odometry threads as well as the back-end optimization. On a PC with an Intel Core i5-9400 2.9 GHz CPU, 16 GB RAM, our orientation detection method has an average time cost of 9 ms per object in each frame, because only a 4×4 matrix multiplication and inverse calculation are required. This only brings an additional 29% time cost and the front-end tracking can still run in real time. The back-end map optimization occurs when a new keyframe with object bounding box is created; the average time of back-end optimization is 293 ms within the four existing object landmarks in the factor graph, with three of them considered to be symmetric in the ICL_home dataset.

5. Conclusions

In this paper, we propose a method to utilize the symmetry of human-made objects to enhance landmark estimation for the monocular object-SLAM system. Based on our light-weight single-frame symmetry-detection technique, we add symmetry factors in the back-end graph of object-SLAM, which significantly improves the object-mapping accuracy according to experiments. The symmetry constraints on the object orientation provide more information for semantic navigation and help the estimation of the scale and center of the objects. Our proposed model lacks competitiveness with other task-specific SLAM models for the specific task, such as the cube-SLAM on ideal cubic objects. Moreover, the proposed method is based on a general geometric model, which is purely model-driven and lacks scalability to data with a more complicated structure or noisy data. Therefore, the combination of the current SLAM model with data-driven machine learning methods may improve the performance of SLAM and substantially reduce the mapping error.

Considering future work, it will be promising to further explore the properties of objects to help robots to better, or even actively, build an object-level map.

Author Contributions: The experimental model was built by W.W. Analyses were carried out by Z.F. and J.H. The organization of data was led by J.H. The descriptions of text use were led by Z.F. All authors have read and agreed to the published version of the manuscript.

Funding: Thiswork was supported, in part, by the National Key Research and Development Program of China under Grant 2020YFB1313600, and, in part, by the National Key Research and Development Program of China under Grant 2022YFB4400402.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets were derived from TUM RGB-D (https://vision.in.tum.de/data/datasets/rgbd-dataset (accessed on 1 January 2022)) and ICL-NUIM (https://www.doc.ic.ac.uk/~ahanda/VaFRIC/iclnuim.html (accessed on 1 January 2022)).

Acknowledgments: The authors thank Yutong Hu for his assistance.

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

References

1. Javaid, M.; Haleem, A.; Singh, R.P.; Suman, R. Substantial capabilities of robotics in enhancing industry 4.0 implementation. *Cogn. Robot.* **2021**, *1*, 58–75. [CrossRef]
2. Shan, T.; Englot, B. Lego-Loam: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765.
3. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-Time Loop Closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278. [CrossRef]
4. Mur-Artal, R.; Tardós, J.D. Orb-Slam2: An Open-Source Slam System for Monocular, Stereo, and Rgb-d Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
5. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
6. Garg, S.; Sunderhauf, N.; Dayoub, F.; Morrison, D.; Cosgun, A.; Carneiro, G.; Wu, Q.; Chin, T.J.; Reid, I.; Gould, S.; et al. Semantics for Robotic Mapping, Perception and Interaction: A Survey. *Found. Trends Robot.* **2020**, *8*, 1–224. [CrossRef]
7. Runz, M.; Buffier, M.; Agapito, L. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 16–20 October 2018; pp. 10–20. [CrossRef]
8. Wu, Y.; Zhang, Y.; Zhu, D.; Feng, Y.; Coleman, S.; Kerr, D. EAO-SLAM: Monocular Semi-Dense Object SLAM Based on Ensemble Data Association. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2021; pp. 4966–4973.
9. Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; Urtasun, R. Monocular 3d Object Detection for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2147–2156.
10. Yang, S.; Scherer, S. CubeSLAM: Monocular 3D Object SLAM. *IEEE Trans. Robot.* **2019**, *35*, 925–938.
11. Nicholson, L.; Milford, M.; Sünderhauf, N. QuadricSLAM: Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1–8. [CrossRef]
12. Hosseinzadeh, M.; Latif, Y.; Pham, T.; Suenderhauf, N.; Reid, I. Structure Aware SLAM Using Quadrics and Planes. In *Proceedings of the Computer Vision—ACCV 2018*; Jawahar, C.V., Li, H., Mori, G., Schindler, K., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; pp. 410–426. [CrossRef]
13. Ok, K.; Liu, K.; Frey, K.; How, J.P.; Roy, N. Robust Object-based SLAM for High-speed Autonomous Navigation. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 669–675. [CrossRef]
14. Jablonsky, N.; Milford, M.; Sünderhauf, N. An Orientation Factor for Object-Oriented SLAM. *arXiv* **2018**, arXiv:1809.06977.
15. 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. Autom. Lett.* **2022**, *7*, 4008–4015. [CrossRef]
16. Roy, A.M.; Bhaduri, J.; Kumar, T.; Raj, K. WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection. *Ecol. Inform.* **2022**, 101919. [CrossRef]
17. Chandio, A.; Gui, G.; Kumar, T.; Ullah, I.; Ranjbarzadeh, R.; Roy, A.M.; Hussain, A.; Shen, Y. Precise single-stage detector. *arXiv* **2022**, arXiv:2210.04252.
18. Jiang, B.; Chen, S.; Wang, B.; Luo, B. MGLNN: Semi-supervised learning via multiple graph cooperative learning neural networks. *Neural Netw.* **2022**, *153*, 204–214. [CrossRef] [PubMed]

19. Salas-Moreno, R.F.; Newcombe, R.A.; Strasdat, H.; Kelly, P.H.; Davison, A.J. Slam++: Simultaneous Localisation and Mapping at the Level of Objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1352–1359.
20. Chen, S.; Song, S.; Zhao, J.; Feng, T.; Ye, C.; Xiong, L.; Li, D. Robust Dual Quadric Initialization for Forward-Translating Camera Movements. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4712–4719. [CrossRef]
21. Tschopp, F.; Nieto, J.; Siegwart, R.; Cadena Lerma, C.D. *Superquadric Object Representation for Optimization-Based Semantic SLAM*; Working Paper; ETH Zurich, Autonomous System Lab: Zurich, Switzerland, 2021. [CrossRef]
22. Sucar, E.; Wada, K.; Davison, A. NodeSLAM: Neural Object Descriptors for Multi-View Shape Reconstruction. In Proceedings of the 2020 International Conference on 3D Vision (3DV), Fukuoka, Japan, 25–28 November 2020; pp. 949–958. [CrossRef]
23. Shan, M.; Feng, Q.; Jau, Y.Y.; Atanasov, N. ELLIPSDF: Joint Object Pose and Shape Optimization with a Bi-Level Ellipsoid and Signed Distance Function Description. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5946–5955.
24. Speciale, P.; Oswald, M.R.; Cohen, A.; Pollefeys, M. A Symmetry Prior for Convex Variational 3d Reconstruction. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 313–328.
25. Srinivasan, N.; Dellaert, F. An Image-Based Approach for 3D Reconstruction of Urban Scenes Using Architectural Symmetries. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 362–370. [CrossRef]
26. Liu, J.; Slota, G.; Zheng, G.; Wu, Z.; Park, M.; Lee, S.; Rauschert, I.; Liu, Y. Symmetry Detection from RealWorld Images Competition 2013: Summary and Results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Portland, OR, USA, 23–28 June 2013.
27. Funk, C.; Lee, S.; Oswald, M.R.; Tsogkas, S.; Shen, W.; Cohen, A.; Dickinson, S.; Liu, Y. 2017 ICCV Challenge: Detecting Symmetry in the Wild. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 1692–1701. [CrossRef]
28. Loy, G.; Eklundh, J.O. Detecting Symmetry and Symmetric Constellations of Features. In *Proceedings of the Computer Vision—ECCV 2006*; Leonardis, A., Bischof, H., Pinz, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 508–521.
29. Cicconet, M.; Hildebrand, D.G.; Elliott, H. Finding Mirror Symmetry via Registration and Optimal Symmetric Pairwise Assignment of Curves. In Proceedings of the ICCV Workshops, Venice, Italy, 22–29 October 2017; pp. 1749–1758.
30. Patraucean, V.; Grompone von Gioi, R.; Ovsjanikov, M. Detection of Mirror-Symmetric Image Patches. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Portland, OR, USA, 23–28 June 2013.
31. Atadjanov, I.; Lee, S. Bilateral symmetry detection based on scale invariant structure feature. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3447–3451. [CrossRef]
32. Atadjanov, I.R.; Lee, S. Reflection Symmetry Detection via Appearance of Structure Descriptor. In *Proceedings of the Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 3–18.
33. Elawady, M.; Ducottet, C.; Alata, O.; Barat, C.; Colantoni, P. Wavelet-Based Reflection Symmetry Detection via Textural and Color Histograms: Algorithm and Results. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops, Venice, Italy, 22–29 October 2017.
34. Gnutti, A.; Guerrini, F.; Leonardi, R. Combining Appearance and Gradient Information for Image Symmetry Detection. *IEEE Trans. Image Process.* **2021**, *30*, 5708–5723. [CrossRef] [PubMed]
35. Doherty, K.J.; Baxter, D.P.; Schneeweiss, E.; Leonard, J.J. Probabilistic Data Association via Mixture Models for Robust Semantic SLAM. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 1098–1104. [CrossRef]
36. Zhang, J.; Yuan, L.; Ran, T.; Tao, Q.; He, L. Bayesian Nonparametric Object Association for Semantic SLAM. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5493–5500. [CrossRef]
37. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97.
38. Sturm, J.; Burgard, W.; Cremers, D. Evaluating Egomotion and Structure-from-Motion Approaches Using the TUM RGB-D Benchmark. In Proceedings of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS), Vilamoura, Portugal, 7–12 October 2012.
39. Handa, A.; Whelan, T.; McDonald, J.; Davison, A.J. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1524–1531. [CrossRef]
40. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.