sensors

MDPI

Article

Mobile Manipulation Integrating Enhanced AMCL High-Precision Location and Dynamic Tracking Grasp

Huaidong Zhou [1], Wusheng Chou [1,2,*], Wanchen Tuo [1], Yongfeng Rong [1] and Song Xu [1]

[1] Robotics Institute, School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China; hdzhou@buaa.edu.cn (H.Z.); tuowanchen@buaa.edu.cn (W.T.); rongyongfeng2014@126.com (Y.R.); keithxs@buaa.edu.cn (S.X.)

[2] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China

* Correspondence: wschou@buaa.edu.cn

Received: 17 October 2020; Accepted: 19 November 2020; Published: 23 November 2020

check for updates

Abstract: Mobile manipulation, which has more flexibility than fixed-base manipulation, has always been an important topic in the field of robotics. However, for sophisticated operation in complex environments, efficient localization and dynamic tracking grasp still face enormous challenges. To address these challenges, this paper proposes a mobile manipulation method integrating laser-reflector-enhanced adaptive Monte Carlo localization (AMCL) algorithm and a dynamic tracking and grasping algorithm. First, by fusing the information of laser-reflector landmarks to adjust the weight of particles in AMCL, the localization accuracy of mobile platforms can be improved. Second, deep-learning-based multiple-object detection and visual servo are exploited to efficiently track and grasp dynamic objects. Then, a mobile manipulation system integrating the above two algorithms into a robotic with a 6-degrees-of-freedom (DOF) operation arm is implemented in an indoor environment. Technical components, including localization, multiple-object detection, dynamic tracking grasp, and the integrated system, are all verified in real-world scenarios. Experimental results demonstrate the efficacy and superiority of our method.

Keywords: mobile manipulation; object detection; visual servo; dynamic tracking grasp; triangle match

1. Introduction

Mobile manipulation is an important issue in the field of robotics. Traditionally, though the breaking down of sophisticated tasks into many single tasks, fixed-base manipulators, such as palletizing robots [1] and assembly robots [2], can complete tasks with high precision via pre-programming. Emerging applications in the industry, such as material handling and mobile welding, demand high mobility and flexibility of robots. A considerable amount of literature has grown up around the theme of mobile manipulation in recent decades [3].

With the developments of robotic technology and artificial intelligence, mobile manipulation has become increasingly practical. In a typical mobile manipulating process, the mobile manipulator should autonomously and safely reach the operating space, avoiding obstacles and achieving self-localization during movement. Then, it will proactively detect and locate the target and complete the grasping task. On the one hand, external sensors (e.g., laser radars) and simultaneous localization and mapping (SLAM) algorithms facilitate the autonomous navigation and localization of mobile robots [4]. The adaptive Monte Carlo localization (AMCL) algorithm has been proved to be an efficient probabilistic localization method [5]. On the other hand, with the assistance of deep learning, more environmental information can be extracted from the vision sensors, and the robot's ability to perceive the environment is enhanced [6]; thus, the accuracy of object detection has been greatly improved. Accordingly, mobile manipulation can provide highly flexible services and dexterous operations in structured and unstructured environments.

Autonomous mobile manipulators are gradually used in many fields, such as various industries [7], living spaces [8–10], and agriculture [11,12].

However, some complex tasks pose huge challenges for mobile manipulation. First, high-precision localization is critical for sophisticated work such as fixed-point operation and material handling in a warehouse environment. Existing approaches need to further improve the localization accuracy and cannot handle the kidnapping problem well, especially in highly similar environments. Furthermore, in dynamic scenarios (e.g., an assembly line), the objects to be grasped are usually diverse and moving. How to rapidly locate and accurately grasp these dynamic objects based on visual information is another important issue, which remains to be further studied [13].

To address the above problems, in this paper, we present a mobile manipulation system capable of performing high-precision localization and dynamic tracking grasp. To accomplish this task, we decompose the system into two subsystems: the navigation subsystem and the grasp subsystem. By using triangle-matched laser-reflector landmarks to continuously improve the weight and accuracy of the particles in AMCL, the navigation subsystem realizes rapid navigation toward the target area and achieves high-precision positioning. The grasp subsystem dynamically grasps moving objects by visual servo, which uses a deep-learning model for real-time object detection and uses the closed-loop feedback of the image position to achieve dynamic object tracking. The main contributions of this work are provided as follows:

(1) The localization accuracy of the AMCL algorithm is improved by adaptively adjusting the weight of particles based on the laser-reflector information.
(2) Efficient dynamic grasping is realized by exploiting real-time object-detection results to achieve closed-loop tracking.
(3) An integrated system containing hardware and software is developed and tested in real-world environments, which is capable of autonomous high-precision localization and dynamic object grasping. The key algorithm components and the complete system are both verified.

The paper is organized as follows. In Section 2, related work on mobile manipulation is reviewed. Section 3 presents the improved localization algorithm and the dynamic grasping policy. In Section 4, real-world experiments of mobile manipulation are described and discussed. Section 5 draws the conclusions and future work.

2. Related Work

Mobile manipulation is a fundamental problem in the field of robotics, and researchers are conducting extensive research on this problem. In the following section, we discuss and review some of the work related to mobile manipulation and our methods.

2.1. Mobile Manipulation

Extensive research on mobile manipulation has been conducted in many robot subdivisions, such as humanoid robots [14,15], wheeled robots [7,16], legged robots [17,18], unmanned aerial vehicles [19–21] and underwater robots [22]. In recent years, a large number of competitions related to mobile operations have been held, such as the Amazon Robotics Challenge [23,24], the DARPA Robotics Challenge [25], and the FetchIt! Mobile Manipulation Challenge [26]. The results of these challenges and projects suggest that the research and application of mobile manipulation is a logical integration of operation, navigation, perception, self-learning and multi-robot collaboration; therefore, we would expect that a mobile manipulators should be able to perform complex tasks in both structured and unstructured environments.

Many mobile manipulators are designed to handle complex tasks in different scenarios. Human Support Robot (HSR) [27] is a real home robot that can be used in a real home environment. The robot can accomplish the Tidy-Up task independently according to the layout of the furniture and the environment information given in advance. In [28], in order to overcome the system uncertainties and anomalies inherent in autonomous mobile manipulator, a coordinated control strategy combining visual and force servos into a sophisticated reactive task control is proposed, and this control strategy achieved high reliability in the "peg-in-hole" type of insertion assembly task. To enhance the autonomy of mobile manipulation, Chen et al. [29] developed a stereoscopic vision system based on the modified iteration of the closest point algorithm to optimize the point cloud and enhance the teaching ability of the robot by using the stereoscopic vision-guided tele-operation control algorithm. Through these two functions, the mobile manipulator can learn semi-autonomously and work autonomously. In 2019, Chen et al. [30] improved the stereo vision algorithm and the iterative closest point (ICP) algorithm for object pose estimation. Based on this algorithm, the robot can select and adjust the robot's pose, and the grasping pose relies on the object pose to maximize the maneuverability of the end-effector. With the development of deep learning, it is constantly applied to mobile operations. In reference [31], Wang et al. proposed a novel mobile manipulation system, which combines the deep reinforcement learning algorithm with visual perception. Although using the visual perception advantage of deep reinforcement learning, the robot can operate the grasping task in an unstructured environment. The effectiveness of the mobile manipulation system is verified in simulations and in the real world.

2.2. *SLAM for Manipulation*

The navigation and location of mobile robots in uncertain environments are always the basic subjects of robot research. A lot of research and reviews have been undertaken to explore the simultaneous localization and mapping (SLAM) in mobile manipulation [6,32–34]. Kohlbrecher et al. [35] described a Hector-SLAM-based [36] open-source SLAM system for urban search-and-rescue missions, which enables robots to map and locate themselves in a degraded urban environment, and independently explore disaster sites to identify victims and other interesting objects. Similarly, Wang et al. [37] used optical flow calculation and wavelet transformation to solve the problem of SLAM in emergency search-and-rescue environments. The three-dimensional (3D) reconstruction, based on RGB-D information, is carried out by combining a two-dimensional (2D) map and three-dimensional objects in order to realize the positioning of the robot in unfamiliar environments. In [38], Xu et al. proposed a mobile robot extensible positioning system. The system integrates a stereoscopic camera, an inertial measurement unit (IMU) and an ultra-ultra-wideband network made up of five anchors. Based on the mobile robot constantly adding new anchor points to the network and estimating anchor point location by recursive least square (RLS), the network is extended. With extended networks, the robot can determine its location in unfamiliar environments. For dynamic environments, the work of [39] presents a SLAM system that automatically projects objects into dynamic environments for semantic processing and creates a global 3D environment map to eliminate the influence of dynamic objects and unknown objects.

2.3. *Visual Tracking Grasp*

Dynamic tracking grasp is the most challenging research area in mobile manipulation. Despite the excellent results of the fixed-based grasp technique, the mobile-based grasp still faces great challenges, and researchers have conducted numerous studies on the subject. The key to dynamic tracking grasp is target tracking, especially visual-based target tracking. Zhang et al. [40] detected a target using the support vector machine (SVM) classifier, which was trained by the scale-invariant feature transform (SIFT) feature points, and used the Kalman filter to estimate the location and direction of the target. This method captures objects on the conveyor belt at a speed of 0.4 m/s. Similarly, Kim [41] developed an Opti-Track system to track the target ball, load by objects, and quickly predict the trajectory of the target and capture it, where the trajectory of the target was learned and trained by machine-learning algorithms. Furthermore, in [42], the authors proposed a method of dynamic switching between local and global planners to track and grasp moving objects. The local planner maintained the end-effector with a steady grasping posture to smoothly follow the object motion. If the previous graspable part becomes unreachable, the global planner quickly moves the end-effector to a new grasp pose to track and grasp the target. In contrast to the previous methods for tracking fast-moving objects, Ardon et al. [43] designed a fully functional grasping system based on the visual servo framework for humanoid service robots to track and grasp targets. This system uses a marker-less tracker model to track the target, and deal with the inaccurate kinematics of the robot based on a pattern tracking method for the end-effector. All of these are based on monocular camera and improve the grasp success rate by 48.8%. In [44], the authors also designed a precision assembly robot system for small components. This system adopts monocular vision guidance for dynamic grasping and two eye-to-hand cameras are used to posture alignment, where the end-effector position is compensated for according to differential motion principle.

3. Method

To accomplish autonomous mobile operation in a complex environment, the robot needs to reach the operating area quickly and accurately, then search and lock the target, and complete the operating task. Therefore, the mobile manipulation system needs to be composed of different subsystems, such as mobile chassis, manipulation arm, navigation and positioning systems, and environment perception and object-detection systems. To elaborate on the method we propose in this paper, the mobile manipulation system is divided into the SLAM subsystem, including the mobile chassis and navigation system, and the grasping subsystem, including the manipulation arm and perception system.

3.1. Enhanced Localization Based on Laser-Reflector Landmarks

Location plays an important role in mobile manipulation. In order to further improve the positioning accuracy of the mobile platform, we improved the particle filter positioning algorithm in AMCL based on the triangle matching. This idea was inspired by the method proposed by de Miguel et al. in [45]. Several modifications are made to integrate reflector information and the result of trilateration into the loop. In this section, all those modifications and adjustments are described in detail.

3.1.1. Map Building and Matching Template Library Establishment

The autonomous movement of the robot depends on reusable environmental maps. For our method, we use Gmapping [46] to build the occupancy grid map of the scenarios and mark the position of the laser reflector as a landmark point as shown in Figure 1a. We establish the global triangle-matching template library **M**, which contains all the triangles composed of any three landmarks in the map. Each item of **M** contains the side length and the vertex coordinates of the triangle. All items are sorted by the maximum side length of corresponding triangles, as shown in Figure 1b.

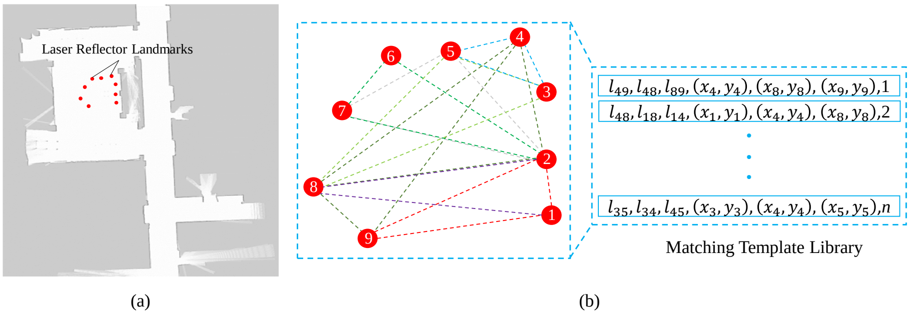
Laser Reflector Landmarks
6
5
4
3
7
2
8
1
9
$l_{49}, l_{48}, l_{89}, (x_4, y_4), (x_8, y_8), (x_9, y_9)$,1
$l_{48}, l_{18}, l_{14}, (x_1, y_1), (x_4, y_4), (x_8, y_8)$,2
$l_{35}, l_{34}, l_{45}, (x_3, y_3), (x_4, y_4), (x_5, y_5)$,$n$
Matching Template Library
(a)
(b)

Figure 1. Map building and template library establishment. (**a**) Map generated by Gmapping [46], where the red dots are laser-reflector landmarks. (**b**) Establishing the matching template library **M** of the laser-reflector landmarks. In the template library, all items contain the side length and the vertex coordinates of the triangles and are sorted by the maximum side length.

3.1.2. Laser-Reflector Identification Based on Triangle Matching

The mobile robot uses the detected laser reflectors to position itself, and its positioning accuracy depends on the position accuracy of the detected laser reflectors. When the robot detects the laser-reflector landmarks, it will identify their ID and obtain corresponding coordinates through triangle matching. If the number of detected laser reflectors is no less than 3, the laser-reflector identification process based on triangle matching will be activated. First, based on the received laser data (distance and orientation information), the mobile robot calculates the relative locations of the detected laser reflectors, and thus obtains the side length of the detected triangles. Then, the mobile robot searches the matched triangle in the matching template library by the side length of the detected triangle. Once all side lengths of the detected triangle match successfully with a matched triangle in the library, its coordinate is identified and added to the set of the identified laser reflectors **R**. Taking Figure 2 as a simple example, A, B and C are the detected laser reflectors and their coordinates are (x_A, y_A), (x_B, y_B) and (x_C, y_C), respectively. The detected triangle $\triangle ABC$ formed by these three detected laser reflectors has side lengths of l_{AB}, l_{AC}, and l_{BC}. Assuming $l_{AB} > l_{AC} > l_{BC}$, the maximum length l_{AB} is first searched in **M**, and l_{AC}, l_{BC} are used to match the triangle sequentially. As a result, the detected triangle $\triangle ABC$ matches the triangle $\triangle 289$ in the library. In this case, A, B and C would be identified as 2, 8 and 9, respectively.

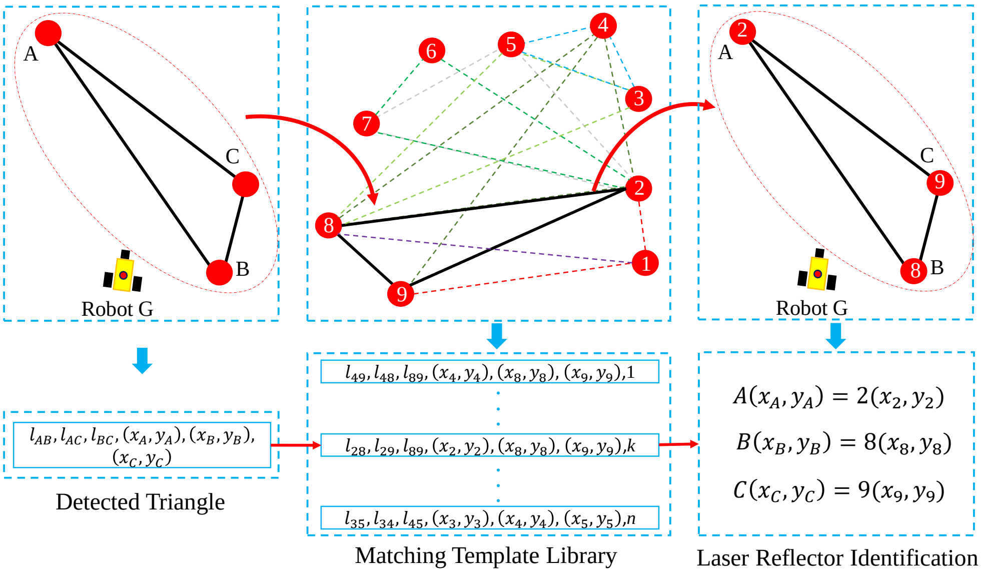
A
C
B
Robot G
6
5
4
3
7
2
8
1
9
2
A
C
9
8
B
Robot G
$l_{AB}, l_{AC}, l_{BC}, (x_A, y_A), (x_B, y_B), (x_C, y_C)$
Detected Triangle
$l_{49}, l_{48}, l_{89}, (x_4, y_4), (x_8, y_8), (x_9, y_9), 1$
$l_{28}, l_{29}, l_{89}, (x_2, y_2), (x_8, y_8), (x_9, y_9), k$
$l_{35}, l_{34}, l_{45}, (x_3, y_3), (x_4, y_4), (x_5, y_5), n$
Matching Template Library
$A(x_A, y_A) = 2(x_2, y_2)$
$B(x_B, y_B) = 8(x_8, y_8)$
$C(x_C, y_C) = 9(x_9, y_9)$
Laser Reflector Identification

Figure 2. The process of triangle matching. A, B, C are the detected laser reflectors and (x_i, y_i) is the coordinate of the i-th detected laser reflector. l_{ij} is the distance between the reflector i and j. G is the mobile robot. The number is the ID of the laser-reflector landmarks in matching template library **M**.

3.1.3. Position Weight Calculation

AMCL is a robot probability localization method that tracks the robot position by particle filter. In this paper, we change the weight and position of the particles to solve the problem of kidnapping and improve the location accuracy. Unlike [45], we use the estimated position of the mobile robot as input to enhance the weight of the particles in the AMCL algorithm. During navigation, the robot detects many laser reflectors and quickly determines the location of the laser reflector based on the triangle matching. The position of the robot can be estimated by the principle of trilateration.

Assuming the robot detects n laser reflectors, we can obtain the ID and corresponding coordinates of each detected laser reflector by triangle matching. Therefore, the relationship between the robot and the detected laser reflectors can be described as follows:

$$\begin{cases} (x_G - x_1)^2 + (y_G - y_1)^2 = r_1^2 \\ (x_G - x_2)^2 + (y_G - y_2)^2 = r_2^2 \\ \vdots \\ (x_G - x_i)^2 + (y_G - y_i)^2 = r_i^2 \\ \vdots \\ (x_G - x_n)^2 + (y_G - y_n)^2 = r_n^2 \end{cases} \tag{1}$$

where (x_G, y_G) represents the estimation position of the robot which can be obtained by solving Equation (1) with the least squares method. (x_i, y_i) is the position of i-th detected laser reflector. Moreover, the orientation of the robot can be obtained as follows:

$$\theta_G = \begin{cases} \frac{1}{n}\sum\limits_{i=1}^{n}\left[\arctan\left(\frac{y_i - y_G}{x_i - x_G}\right) - \varphi_i\right] & x_i > x_G \\ \frac{1}{n}\sum\limits_{i=1}^{n}\left[\arctan\left(\frac{y_i - y_G}{x_i - x_G}\right) - \varphi_i + \pi\right] & y_i \geq y_G, x_i < x_G \\ \frac{1}{n}\sum\limits_{i=1}^{n}\left[\arctan\left(\frac{y_i - y_G}{x_i - x_G}\right) - \varphi_i - \pi\right] & y_i < y_G, x_i < x_G \\ \frac{1}{n}\sum\limits_{i=1}^{n}\left[\frac{\pi}{2} - \varphi_i\right] & y_i > y_G, x_i = x_G \\ \frac{1}{n}\sum\limits_{i=1}^{n}\left[-\frac{\pi}{2} - \varphi_i\right] & y_i < y_G, x_i = x_G \end{cases} \tag{2}$$

where θ_G represents the orientation of the robot, φ is the observation angle of the detected laser reflectors. Therefore, the estimated pose of the robot can be expressed as $\mu_G = (x_G, y_G, \theta_G)$. Figure 3 shows an example of the positioning process where the mobile robot G detects four reflectors A, B, C and D. The estimated position of the robot can be obtained by Equations (1) and (2). And the position weight of the particle can be calculated as follows:

$$d^{[i]} = \frac{1}{(2\pi)^{\frac{3}{2}}} \exp\left(-\frac{1}{2}(x_t^{[i]} - \mu_G)^T (x_t^{[i]} - \mu_G)\right) \tag{3}$$

where the position and orientation of each particle is defined as $x_t^{[i]} = (x_i, y_i, \varphi_i)$.

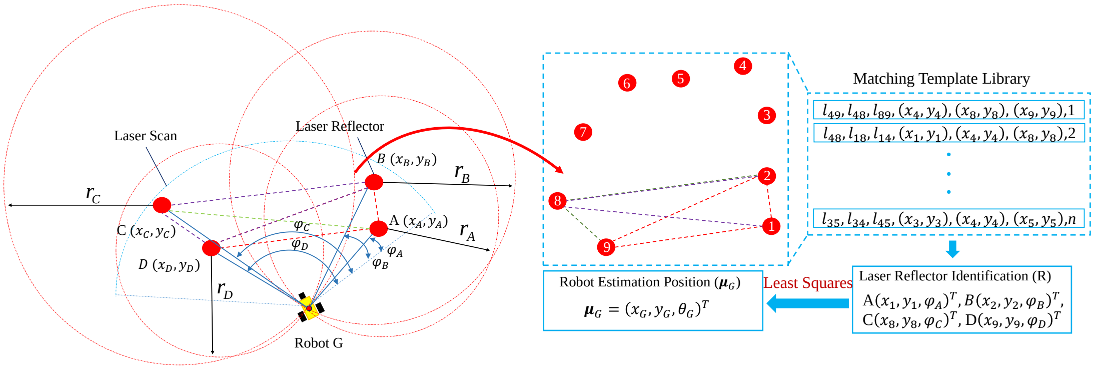
Laser Scan
Laser Reflector
$B\ (x_B, y_B)$
r_B
r_C
$A\ (x_A, y_A)$
r_A
$C\ (x_C, y_C)$
φ_C
φ_D
φ_A
φ_B
$D\ (x_D, y_D)$
r_D
Robot G
6
5
4
3
7
2
8
1
9
Matching Template Library
$l_{49}, l_{48}, l_{89}, (x_4, y_4), (x_8, y_8), (x_9, y_9), 1$
$l_{48}, l_{18}, l_{14}, (x_1, y_1), (x_4, y_4), (x_8, y_8), 2$
$l_{35}, l_{34}, l_{45}, (x_3, y_3), (x_4, y_4), (x_5, y_5), n$
Robot Estimation Position ($\boldsymbol{\mu}_G$)
$\boldsymbol{\mu}_G = (x_G, y_G, \theta_G)^T$
Least Squares
Laser Reflector Identification (R)
$A(x_1, y_1, \varphi_A)^T, B(x_2, y_2, \varphi_B)^T,$
$C(x_8, y_8, \varphi_C)^T, D(x_9, y_9, \varphi_D)^T$

Figure 3. The process of positioning. A, B, C and D are the detected laser reflectors. φ and r are the observation angle and detection distance of the detected laser reflector, respectively. G is the mobile robot. θ_G is the orientation of the robot based on the observation angle of the detected laser reflectors. μ_G is the estimation position of the robot.

3.1.4. Orientation Weight Calculation

In navigation, the accuracy of maps determines the accuracy of robot navigation. The map error often leads to location failures and the kidnapping problem. To improve the accuracy of map matching in navigation, the laser-reflector information will be used to enhance the weight of particles in the AMCL algorithm. Therefore, we compute the weight score to evaluate the accuracy of the particle matched with the map based on the Gaussian model [47]. The weight score s_i of the particle is calculated as follows:

$$s^{[i]} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(\varphi_i-\mu)^2}{2\sigma^2}} \tag{4}$$

where n is the number of the detected laser reflectors. μ and σ represents the mean value and the standard deviation of the observation angle, respectively. With this expression, we can evaluate the particle position matched with the map and use it to modify the probability of the newly generated particles.

3.1.5. Weight Adjusting and Particle Generation

In the initialization phase, a set of particles $\chi_t = \left\{(x_t^{[i]}, \omega_t^{[i]}), i = 0, 1, \cdots, N\right\}$ is randomly added. From the AMCL algorithm, the current state of the robot is predicted by the motion model $p\,(x_t|x_{t-1}, u_t)$, where x_{t-1} represents the last state of the robot and u_t describes the control information. The predictive probability density function at time t can be expressed as $p\left(z_t, \bar{\mathbf{M}}|x_t^{[i]}, \mathbf{M}\right)$, where z_t and $\bar{\mathbf{M}}$ represent the observation information and the result of trilateration at time t, respectively. According to [47], the weight $\omega_t^{[i]}$ of each particle can be calculated as follows:

$$\omega_t^{[i]} = \omega_{t-1}^{[i]}p\left(z_t, \bar{\mathbf{M}}|x_t^{[i]}, \mathbf{M}\right) \tag{5}$$

In the positioning process, the new weight of each particle can be generated as follows:

$$\omega_{t-new}^{[i]} = \alpha \cdot s^{[i]} \cdot \omega_t^{[i]} + \beta \cdot d^{[i]} \tag{6}$$

The coefficients α and β are both constants and used to balance the position and the orientation. The coefficients α and β are empirically set to 154 and 1, respectively. The weights are then normalized to make the sum of all the weights equal 1.

In contrast to the AMCL algorithm which determines whether new particles are added according to the position of robot and the weight of particles, and randomly distributes the particles in the map, this paper defines a different function to generate new particles X_{t-new} according to the estimated position μ_G and the probability of generating those new particles. The function of the newly generated particles are to follow a normal distribution centered in μ_G.

$$X_{t-new} = \lambda Q + \mu_G \tag{7}$$

where Q is a random vector with distribution N[0, 1]. The coefficients λ is empirically set to 1. In addition, the probability of generating a new particle is:

$$p\,(x_{t-new}|x_t, u_{t-1}, z_t, \bar{\mathbf{M}}) = \begin{cases} d_i, & \text{if } d_i > 0.01 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

In the experiment, we use the AMCL algorithm to initialize the robot. Then, the robot updates the weight of the particles based on the result of the identified laser reflectors and the estimated position of the robot. Finally, we randomly add new particles near the estimated position μ_G according to the weight of $d^{[i]}$, which indicates the distance between the particle position and the estimated position of the robot. This improved algorithm is shown in Algorithm 1.

Algorithm 1 The improved adaptive Monte Carlo localization (AMCL) algorithm.

```
**Input:** control information $u_t$, observation information $z_t$, matched template libraries $\mathbf{M}$;
**Output:** new particles set $\chi_t$;
Initialization: $\mathbf{R} = \bar{\mathbf{M}} = \chi_t = \emptyset$, $\omega_{t-new} = 0$, new particle $x_{i-new} = \emptyset$, cache particle set $\bar{\chi}_t = \emptyset$;
$\mathbf{R}$,$\bar{\mathbf{M}}$ = LaserReflectorIdentification$(z_t, u_t, \mathbf{M})$;
**if** $\bar{\mathbf{M}} = \emptyset$ **then**
    **for** $i$ = 0 *to* $N$ **do**
        $Sample\ x_t^{[i]} \sim p\left(x_t|x_{t-1}^{[i]}, u_t\right)$;//Original AMCL
        $\omega_t^{[i]} \sim measurement\_model(z_t, x_t^{[i]})$;//Original AMCL
        $\bar{\chi}_t = \bar{\chi}_t + \left\langle x_t^{[i]}, \omega_t^{[i]} \right\rangle$;
        $draw\ i\ withprobability \propto \omega_{t-1}^{[i]}$;
        $add\ x_t^{[i]}\ to\ \chi_t$;
    **end for**
**else**
    $x_t$ = motion_model($x_{t-1}, u_t$);
    **for** $i$ = 0 *to* $N$ **do**
        $d^{[i]} = PositionWeightCalculation(\bar{M}, x_t^{[i]})$;
        $s^{[i]} = OrientationWeightCalculation(z_i)$;
        $\omega_{t-new}^{[i]} = \alpha \cdot s^{[i]} \cdot \omega_t^{[i]} + \beta \cdot d^{[i]}$;
        $\bar{\chi}_t = \bar{\chi}_t + \left\langle x_t^{[i]}, \omega_{t-new}^{[i]} \right\rangle$;
    **end for**
    $\omega_{t-new} = normolaizeweight(\omega_{t-new})$;
    **for** $i$ = 0 *to* $N$ **do**
        $p\left(x_{t-new}|x_t, u_{t-1}, z_t, \bar{\mathbf{M}}\right) = probofnewparticle(d^{[i]})$;//function (8)
        **if** $rand() \leq p\left(x_{t-new}|x_t, u_{t-1}, z_t, \bar{\mathbf{M}}\right)$ **then**
            $x_{i-new} = ParticleGeneration(\omega_{t-new})$;
            $add\ x_{i-new}\ to\ \chi_t$;
        **end if**
    **end for**
**end if**
**return** $\chi_t$
```

3.2. Multiple-Object Detection and Dynamic Tracking Grasp

The ultimate goal of mobile manipulation is to operate the target object. Although methods for manipulating stationary objects or movement-known objects have been well studied [40,48], there are still great challenges for manipulating randomly placed and moving objects. Therefore, we use the object-detection model to detect and locate the object, and use the visual servo technology to control the manipulator to perform a tracking grab on the moving object. In the following section, the details regarding the object-detection model and dynamic tracking grasp are described.

3.2.1. Multiple-Object Detection and Location

To manipulate multiple objects, the robot needs to identify the object and locate its position relative to the mobile manipulator. The main challenge for object detection is the possibility of multiple simultaneous object instances. The instances might vary in size, shape, attitude, and appearance. The target detection module should overcome the influence of environment on vision and detect and locate all objects in the image. In this study, we use the single-shot multi-box detector (SSD) approach, proposed by Liu in [49]. The mobile manipulation mission is a real-time task, and we choose the SSD approach since it is renowned for its accuracy and speed performance.

The object-detection module takes RGB images as input and outputs a list of detected targets. Each detection target contains a category label and the pixel coordinates of the bounding box in the image. Since the bounding box envelops the contour of the target, the center of the bounding box is regarded as the center of the object, whereas the longest side of the bounding box is regarded as the graspable edge. To obtain the 3D position of the object relative to the mobile manipulator, the pose of the object relative to the camera can be obtained by converting the pixel coordinates of the object center to the camera coordinate system and the distance is the camera depth value. Therefore, the position of the object relative to the robot can be obtained by coordinate transformation. The coordinate system transformation is shown in Figure 4.

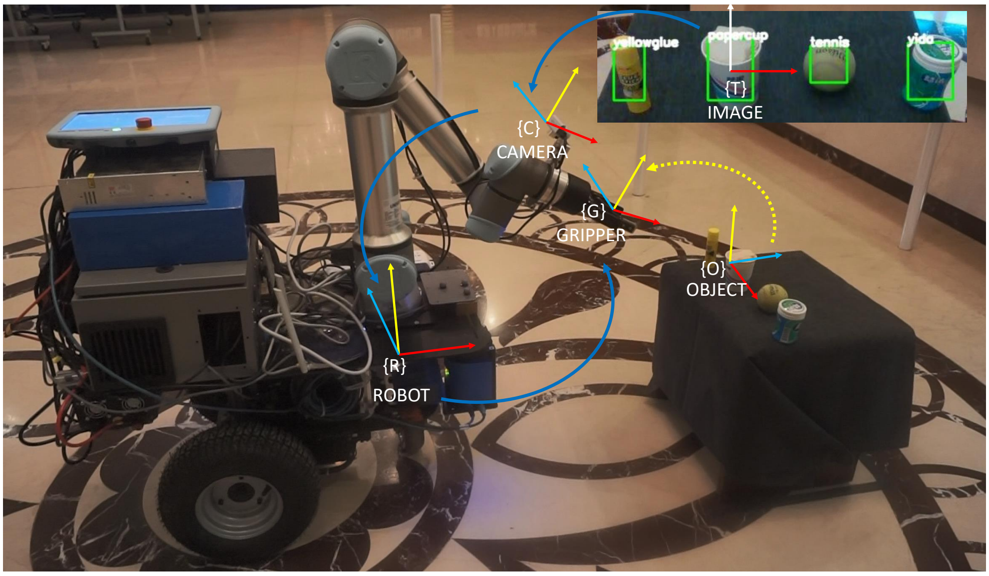

Figure 4. Coordinate transforming from the camera to the robot and gripper.

3.2.2. Dynamic Tracking and Grasping

Grasping is one of the important tasks performed in mobile manipulation, especially the grasping of randomly placed and moving objects. During the process of handling the moving object, the end-effector needs to locate and track the object in real time. In our method, we combine object detection and image-based visual servoing (IBVS) method to track and grasp the objects. From the detection module, taking the image and depth as inputs, we obtained the 2D coordinate in the image and 3D position in the robot coordinate system of the object's center. This means that we can map the end-effector velocities to the velocities of the object center point through the Jacobian matrix, as follows:

$$\dot{p}(t) = J_m(t)\dot{q}(t) \tag{9}$$

where $\dot{p}(t)$ represents the velocity of the object center and $\dot{q}(t)$ represents the end-effector velocity at t time, respectively. $J_m(t)$ is the Jacobian matrix image. Now, we can obtain the desired end-effector velocities, which are converted into the object center velocities via the inverse Jacobi or the pseudo-inverse, as shown in Figure 5.

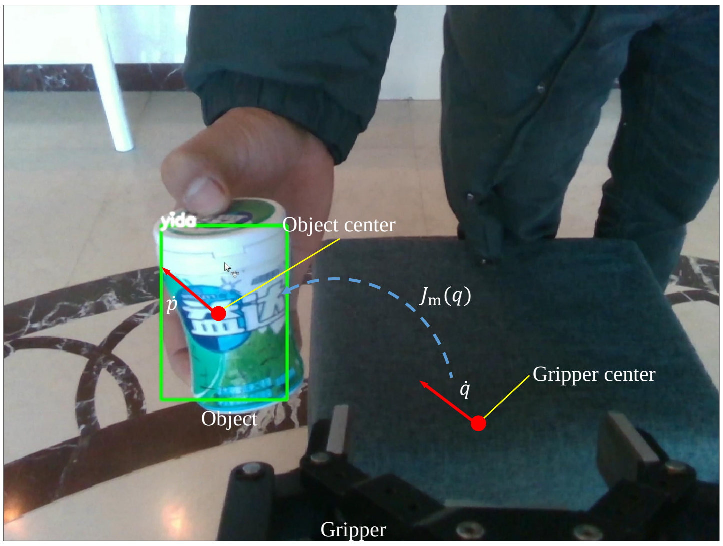

Figure 5. The tracking module based on visual servo.

To track a moving object in real time, we calibrate the gripper center between the fingers of the end-effector as the tracking point. Then, we use the error between the tracking point and the object center as the control parameter. Based on the error, the IBVS method is used as a closed-loop to control the tracking process. The error $e(t)$ between the tracking point and the object center is:

$$e(t) = P_c(t) - P_T(t) \tag{10}$$

where $P_c(t)$ represents the object center point; $P_T(t)$ represents the tracking point. We take the derivative of Function (10) and put the result into Function (9). The velocities of the end-effector can be described as:

$$\dot{q}(t) = -\lambda J_m^+(t)\dot{e}(t) \tag{11}$$

where λ is the parameter of control gain, $J_m^+(t)$ is the pseudo-inverse and represented as follows:

$$J_m^+(t) = \left(J_m^T(t)J_m(t)\right)^{-1} J_m^T(t) \tag{12}$$

Generally, the multi-DOF manipulator has singularity, joint limit and maximum extension limit. To overcome these limitations and achieve better control of the system, we adopt two approaches. First, due to the manipulator being fixed onto the mobile platform, appropriate platform mobility can ensure that the object will be within the manipulator's working space and can avoid the manipulator reaching the limits of its operating position. Second, we reconfigure the mobile manipulator's configuration space in the null space to minimize these undesirable effects. Due to the multiple degrees of freedom, the arms often have a high degree of redundancy and large null spaces in which the system can reconfigure, without changing the pose of the end-effector. To address this, the Function (11) can be further rewritten as:

$$\dot{q}(t) = -\lambda J_m^+(t)\dot{e}(t) + N\dot{q}_0 \tag{13}$$

where N is a null space projection of the Jacobian and $\dot{q}_0$ is a set of additional end-effector velocities. Manipulability is a metric that represents the distance to a manipulator singularity in joint space. In this context, the manipulability is computed as:

$$\Omega = \sqrt{\det(J_m(t)(J_m^T(t))} \tag{14}$$

Please note that when a joint reaches a singular configuration, the Jacobian loses rank and Ω becomes zero due to a zero eigenvalue of $J_m(t)(J_m^T(t))$. We maximize this metric to avoid singularities and improve control stability. In the domain of mobile manipulation, the mobile base can be used to improve manipulability than a manipulator alone. For example, the mobile base can improve the manipulator's maneuverability by moving closer to the target, when the arm extends to the edge of its workspace.

4. Experiment

Mobile manipulation systems usually integrate many actuators and sensors to accomplish complex mobile operation tasks. This section focuses on the evaluation of the proposed mobile manipulation system through experimentation. First, the settings of the mobile manipulator system and the experiment environment are introduced, then the mobile operation task is defined and evaluated.

4.1. System Overview

4.1.1. System Hardware

To evaluate the proposed mobile manipulation system, we developed a mobile manipulator, as shown in Figure 6. The mobile manipulator is composed of a mobile platform, a six-degrees-of-freedom (DOF) arm with a two-finger gripper, and multiple sensors for object detection and environmental perception. The mobile platform consists of a passive universal wheel and two active drive wheels driven by DC motor. The 6-DOF arm is UR5e, supported by Universe Robotics and mounted on the mobile platform, with a workspace of 850 mm and a maximum payload of 5 kg. The end-effector for completing the grasping task is a two-finger gripper, manufactured by Robotiq, with a stroke of 85 mm. The grip force is adjusted to be under 110 N, and the maximal payload is 5kg. The perception sensors include the LIDAR for SLAM and the RGB-D camera used for grasp. The control system of the mobile manipulator consists of an upper control system and a bottom control system. The upper control system runs on a computer equipped with an NVIDIA GPU and is responsible for the navigation control of the mobile platform and the detection and tracking grasp of the objects. The bottom control system runs on an STM32 and is responsible for calculating and executing the motion of each motor according to the upper control movement instruction. The detailed hardware parameters of the mobile manipulator are shown in Table 1.

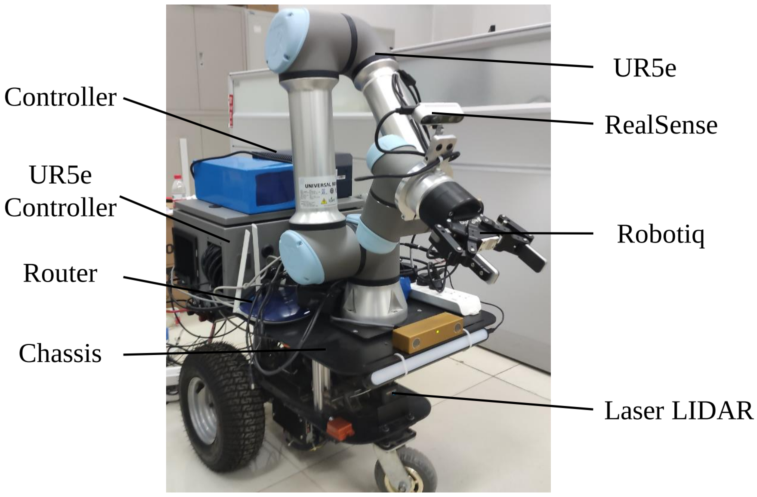

Figure 6. The prototype of the proposed mobile manipulator.

Table 1. The hardware parameters of the mobile manipulator.

Parameter	Value
Computer	I7-9700, 16G, 512SSD
GPU	GTX 1070, NVIDIA
LIDAR	NAV350-3232, SICK
RGB-D Camera	RealSense d435i, Intel
Manipulator	UR5e, UR
Gripper	2F-85 Gripper, Robotiq
Battery	48V 32AH
Chassis size (L × W × H)	810 mm × 493 mm × 671 mm

4.1.2. Control Framework

We use the Robot Operating System (ROS) [50], which integrates the subsystems, such as the mobile chassis, sensors, and manipulator subsystems. The subsystems communicate with each other through the topic. The ROS-Based Control Framework is shown in Figure 7.

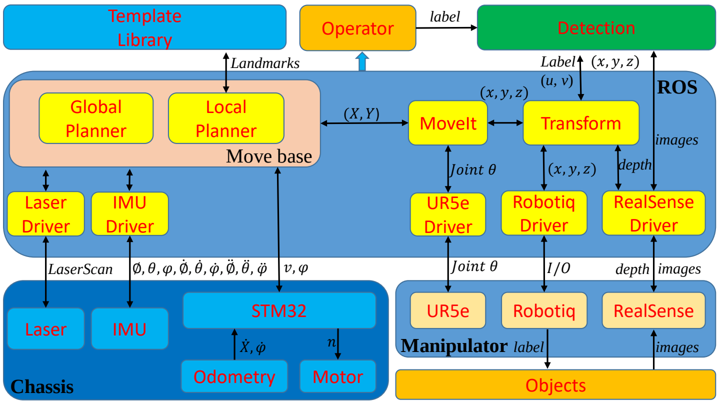

Figure 7. ROS-Based Control Framework.

4.2. *Experiment Setting*

4.2.1. Environment Description

Figure 8 shows the real environment used to evaluate the performance of the proposed mobile manipulation system in practice. The environment includes a lobby and a corridor, as shown in Figure 1. The work area is in the lobby and is surrounded by nine laser reflectors. The target objects are randomly placed on a desk. The height of the desk is 350 mm. The other environment setting is similar to the indoor environment.

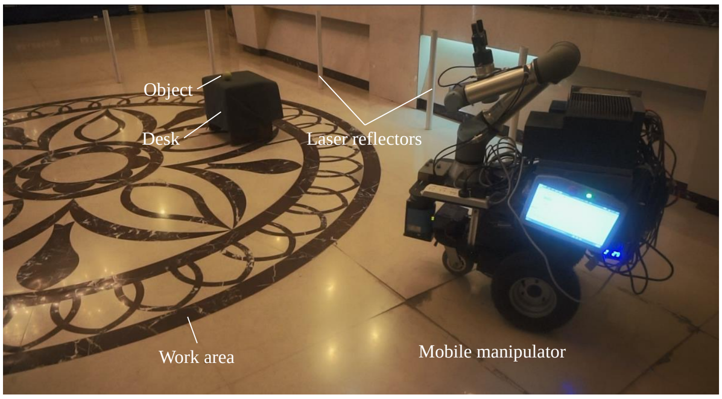

Figure 8. The experimental scenario.

4.2.2. Task Definition

To evaluate the capability of the integrated mobile manipulation system, we choose a classical mobile manipulation task, i.e., a mobile picking task. This manipulation task is challenging because it requires a feasible policy that seamlessly considers both the navigation and manipulation, based on the onboard sensors. The mobile manipulator needs to be able to randomly initialize and start in the corridor, autonomously navigate to the work area, rapidly search the desk, and then accurately detect and grasp the object. The task pipeline is as follows:

(1) The mobile manipulator navigates to the work area without collisions;
(2) The mobile manipulator locates and approaches the desk;
(3) The manipulator detects and grasps the object;
(4) The manipulator dynamically tracks and grasps the moving object.

4.3. Experiment Results

4.3.1. Localization Experiments

In real experiment environments, the location of the robot is randomly initialized. The robot navigates to a target point and measures the error of the position. To evaluate the proposed method, the AMCL [5] and Hector-SLAM [35] algorithms are compared for the position tasks. As shown in Figure 9, the position error provided by our approach is better than the others and has almost the same errors for its orientation. However, compared to the other algorithms, our method has a smaller steady-state error. In 20 repeated experiments, all methods successfully navigated the mobile manipulator to the target point, but the position errors vary between methods. As shown in Table 2, the position error of our proposed approach is lower than the other methods. The position error of the proposed method is within 12 mm and the pose error is within one degree (the angle with x-axis). We consider a position error of less than 12 mm to be the standard for successful navigation, and our proposed method achieves a 95% success rate, which is superior to the other methods. The reason for this is that in the process of mobile manipulator movement, the weight and position of particles are changed based on the triangulation matching, the convergence of particles is accelerated, and thus the position accuracy of the mobile manipulator is improved.

Table 2. The result of the localization

Methods	Position Mean Error (cm)			Pose Mean Error (rad)	Success Rate
	x	y	**Position**	**Theta**	**Position Error < 1.1 cm**
AMCL [5]	3.91	3.32	5.14	0.027	0.20
Hector-SLAM [35]	4.63	3.84	6.12	0.032	0.35
Ours	0.82	0.73	1.09	0.017	0.90

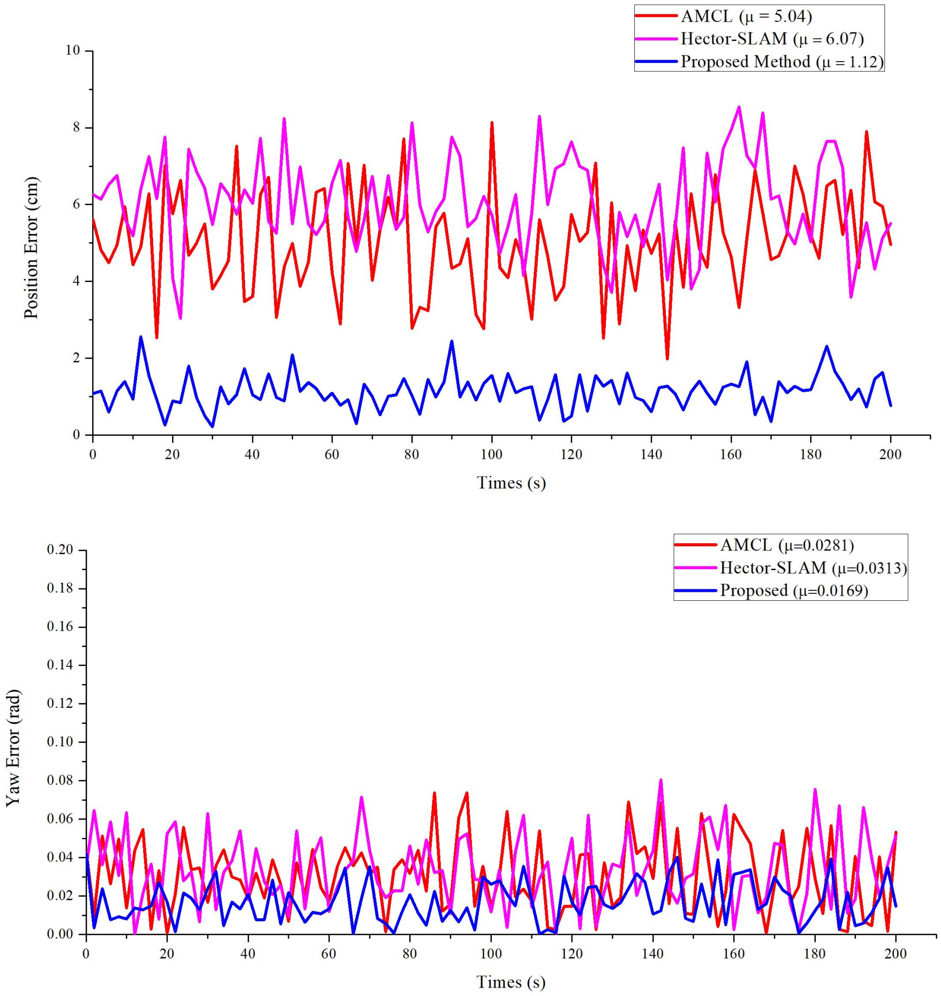

Figure 9. Comparison between the proposed method with AMCL and Hector SLAM. (**a**) Position Error vs. Time, (**b**) Yaw Error vs. Time.

In real scenarios, the situation that less than three reflectors can be detected frequently occurs due to the movement of people or obstacles. To further evaluate the proposed method, AMCL, the trilateration-only method and the method combining trilateration and AMCL (T-AMCL), are compared. As shown in Figure 10, when the robot detects less than three laser reflectors (around 150 s), the trilateration-only method failed and the T-AMCL method took about 25 s to re-initialize and converge, thus reducing the positioning accuracy and robustness. However, our proposed method is to improve the positioning accuracy by improving the particle weights in AMCL. In this case, the particles can be directly applied to the AMCL algorithm and help the robot get out of trouble quickly. Therefore, our proposed method is more robust in dynamic complex environments.

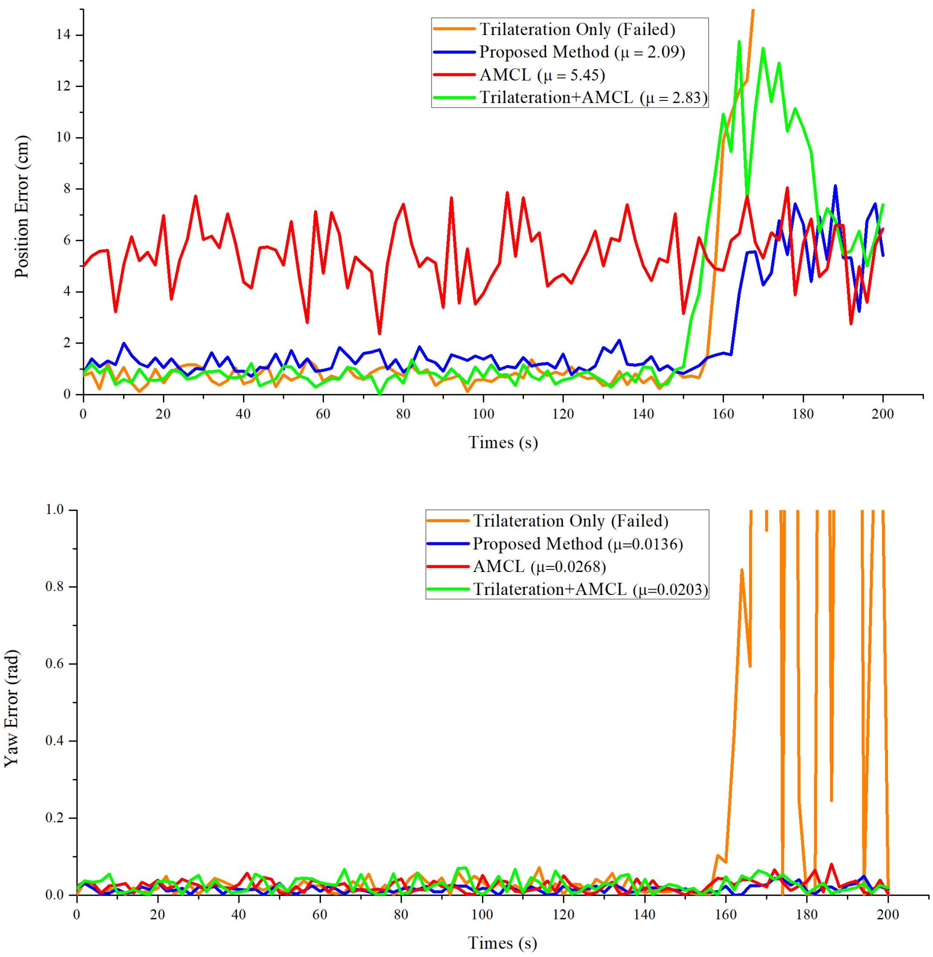

Figure 10. Comparison results of our proposed method with trilateration-only method, AMCL and T-AMCL. (**a**) Position Error vs. Time, (**b**) Yaw Error vs. Time.

4.3.2. Multiple-Object Grasp Experiments

The speed and accuracy of object detection determines the success of mobile manipulation tasks. However, the most time-consuming and memory-consuming part is object detection. To address this issue, we simplify the object-detection model by using the state-of-the-art algorithm NetAdapt [51] to reduce memory consumption. For the trained model, NetAdapt automatically and iteratively identifies and removes redundant channels from the model to reduce the computational complexity. In addition, histogram homogenization, edge detection and mean filtering are used to preprocess the image before being sent into the detection module. This method improves the success rate of target detection in dark environments. In the experiment, we choose three light conditions and seven different objects, such as yida bottle, tennis, paper cup and so on, as shown in Figure 11. For the different objects, the yida bottle and water bottle have a cylinder shape, the yellow glue and the stick are have a slim cylinder shape with different textures, the tennis ball has a ball shape, and the cube has a cube shape. These objects are good representations of most common objects. Table 3 shows the grasp success rates in different light situations with various objects. The result shows that the system achieves a higher success rate, except for the bottle, which is bigger than the others and could lead to unstable grasping. Because blue is a difficult color to be detected in a dark environment, the success rate of detecting the stick is the lowest. The performance of non-reflective objects is better in the high-light environment and comparative in the dark environment compared with the reflective objects. Figure 12 shows the successful detection and grasping for the multiple objects and Figure 13 shows the process of detecting and grasping in the high-light environment. All of these operations can be done in real-time.

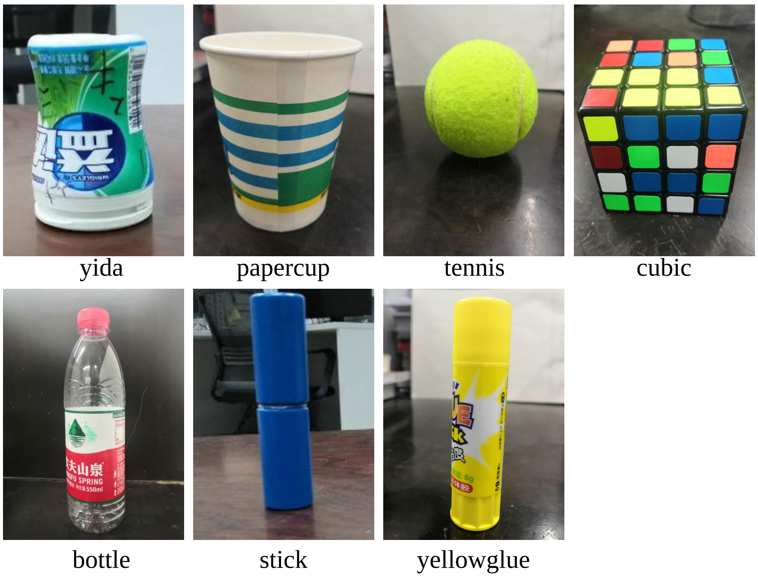

Figure 11. The seven different objects.

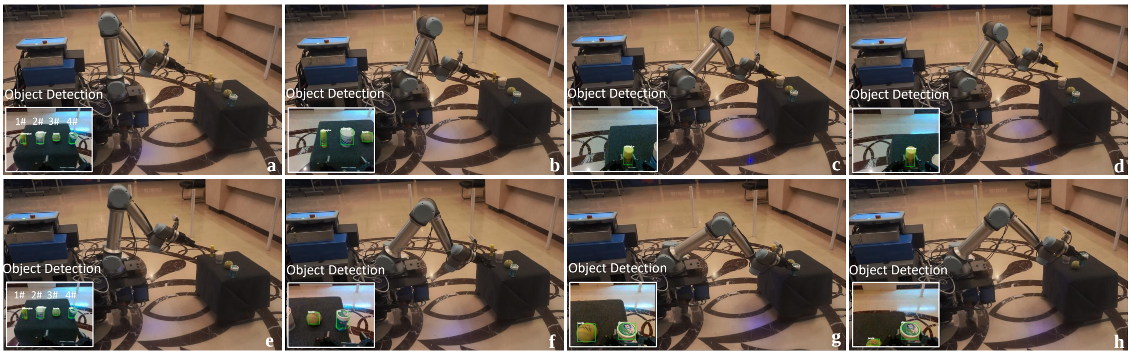

Figure 12. The process of multi-object grasping. (**a**) Detecting the multiple objects(1# yellow glue, 2# paper cup, 3# tennis, 4# yida). (**b**) Approaching the 1# object. (**c**,**d**) Grasping the 1# object. (**e**) Detecting the 4# object. (**f**) Approaching the 4# object. (**g**,**h**) Grasping the 4# object.

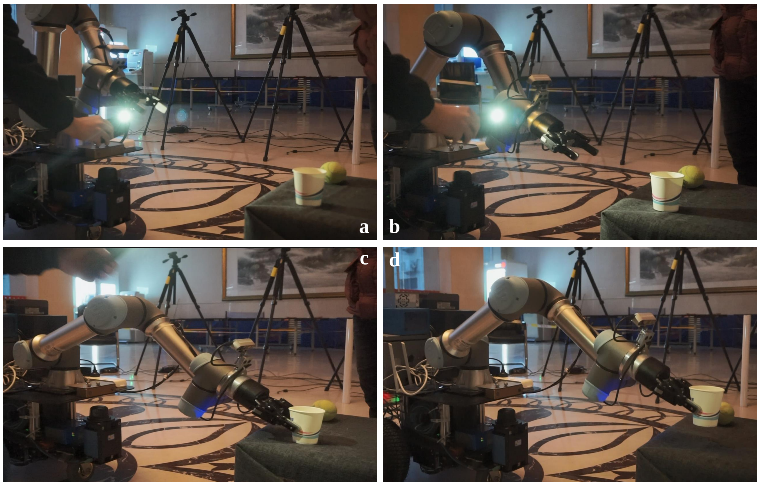

Figure 13. The process of grasping in a high-light environment. (**a**) Detecting the object. (**b**) Approaching the object. (**c**) Grasping the object. (**d**) Placing the object.

Table 3. Mobile manipulation success rate for different objects in different light conditions

Object Type	**Light Conditions**		
	High-Light	**Normal**	**Dark**
yida	0.90	0.95	0.80
yellowglue	0.80	0.90	0.80
tennis	0.95	0.95	0.70
cubic	0.85	0.90	0.75
bottle	0.75	0.85	0.75
stick	0.85	0.90	0.50
papercup	0.90	0.90	0.65

4.3.3. Dynamic Tracking Grasp Experiments

In some situations, the mobile platform needs to deal with a moving object. As such, the manipulator needs to have the ability to dynamically track and grasp the object. As described in Section 3.2, the dynamic tracking grasp systems are combined with the object detection and visual servo. In the tracking grasp experiment, the object is moved randomly by the operator and the manipulator tracking the object depends on the object-detection module. The dynamic tracking grasp process is shown in Figure 14. The result shows that the manipulator can track and grasp the object well when the object velocity is normal ($v < 0.1$ m/s). However, when the object moves fast, the manipulator fails to track the object and stops moving, waiting for the target to reappear in order to continue tracking.

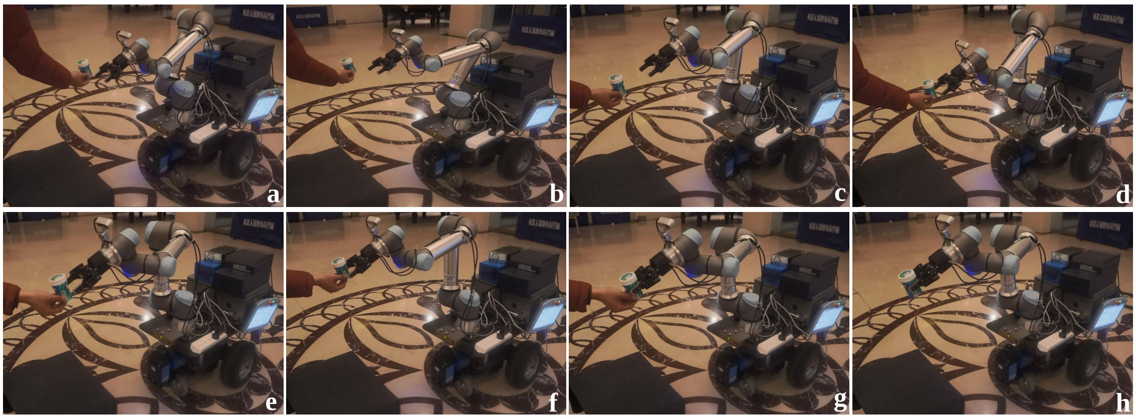

Figure 14. The process of dynamic tracking grasp. (**a**) Detecting the object. (**b**,**c**) Approaching the object. (**d**–**f**) Dynamically tracking the object. (**g**,**h**) Grasping the object.

4.3.4. Integrated System Experiments

To evaluate the integrated system, the mobile manipulator is assigned to execute a complex mobile manipulation task of object grasp. First, the position of the robot is randomly initialized in the corridor. Then, the operator sends the object name. Next, the mobile manipulator navigates to the target area and completes the high-precision self-location and search of the target desk by rotating. Then, the manipulator detects and returns a list of results to the operator and autonomously grabs the target object. Some of the robot navigation and grasping process are shown in Figure 15. Table 4 summarizes the success rates of mobile manipulation with three kinds of objects. Compared with other objects, the papercup has the lowest grasping success rate due to its soft structure, while the tennis ball has the highest detection and grasp success rate, because its spherical shape is easy to be recognized and grasped.

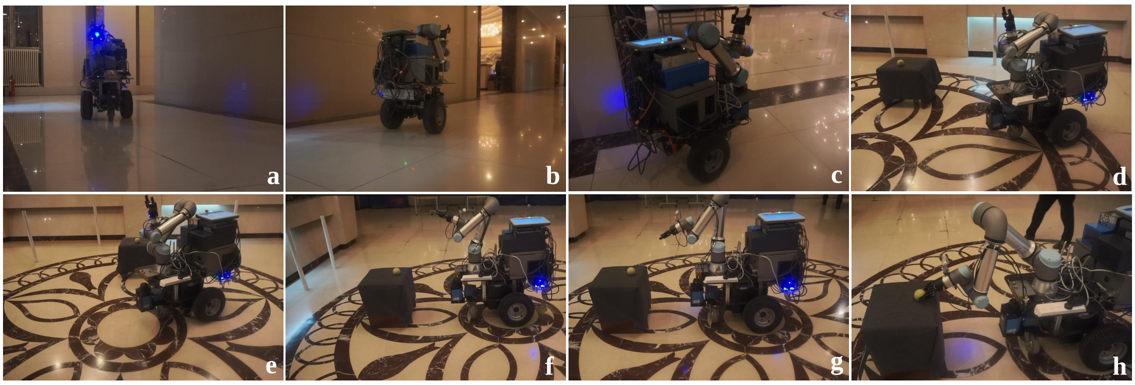

Figure 15. The process of autonomous mobile manipulation. (**a**) Robot initiating. (**b**,**c**) Navigating to the target area. (**d**,**e**) High-precision localization based on laser reflectors. (**f**) Detecting the desk. (**g**) Detecting the object. (**h**) Grasping.

Table 4. Mobile manipulation success rate for different objects (total = 20)

Object Type	**Success Rate**		
	Navigation	**Detection**	**Grasping**
yida	0.85	0.88	0.93
tennis	0.95	0.95	0.94
papercup	0.90	0.89	0.88

4.3.5. Discussion

Our experiments demonstrate the effectiveness of the proposed method on a mobile robot. The mobile manipulator can automatically navigate to the target area and locate itself with high-precision, and then detect and grasp the object based on vision. For the moving object, based on visual servo, it can dynamically track and grasp the object in real time. However, some problems exist. On the one hand, the positioning accuracy depends on the accuracy of the position of the laser reflectors. Although the robot can achieve high-precision positioning based on laser-reflector information, if the position of laser-reflector changes, the matched template library needs to be reconstructed. On the other hand, if we combine object detection with the visual servo, the closed-loop control and dynamic tracking grasp of the manipulator are realized. To grasp the objects precisely, the performance of the proposed method should be improved, which is a subject of our future work.

5. Conclusions

In this paper, a mobile manipulation system is proposed, which integrates navigation, high-precision positioning, object detection and dynamic tracking grasp. Based on the triangle matching with the laser-reflector data, the mobile robot can achieve high-precision positioning in a target area. The object-detection model can detect multiple objects under different lighting conditions. The closed-loop control of manipulator and the dynamic tracking control of moving targets are realized by combining target detection with visual servo. Finally, experiments in the real world demonstrate the effectiveness of the proposed method. The mobile manipulator can autonomously complete mobile manipulation tasks in a real environment, and can achieve a high success rate of 90% for localization and 95% for grasping.

In future research, we will pay more attention to the intelligence and stability of robots in mobile operation. First, we will study the practical application of visual navigation in mobile robots, further integrating visual information with laser data, or replacing laser data to improve the robot's ability to perceive environmental data. Second, we would like to study the object pose estimation method based on vision and apply it to the mobile manipulation. Finally, we will further study the cooperative motion of the mobile chassis and the mechanical arm, and solve the problem of occlusion in the grasping process, and apply it to more complex mobile manipulation tasks and complex environments.

Author Contributions: In this work, H.Z. wrote the manuscript and performed the methodology and experiment; W.C. performed the methodology and reviewed the manuscript; W.T. performed the investigation and analyzed the experiment data; Y.R. performed the software and visualization; S.X. performed the experiments and analyzed the experiment data. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (Grant No. 2019YFB1310802).

Acknowledgments: We would like to thank the anonymous reviewers and academic editor for their comments and suggestions.

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

References

1. Luan, N.; Zhang, H.; Tong, S. Optimum Motion Control of Palletizing Robots Based on Iterative Learning. *Ind. Robot.* **2012**, *39*, 162–168. [CrossRef]
2. Song, J.; Chen, Q.; Li, Z. A Peg-in-hole Robot Assembly System Based on Gauss Mixture Model. *Robot. Comput. Integr. Manuf.* **2021**, *67*, 101996. [CrossRef]
3. Bostelman, R.V.; Hong, T.H.; Marvel, J.A. Survey of Research for Performance Measurement of Mobile Manipulators. *J. Res. Natl. Inst. Stand. Technol.* **2016**, *121*, 342–366. [CrossRef]
4. Oğuz-Ekim, P. TDOA Based Localization and its Application to the Initialization of LiDAR Based Autonomous Robots. *Robot. Auton. Syst.* **2020**, *131*, 103590. [CrossRef]
5. Pfaff, P.; Burgard, W.; Fox, D. Robust Monte-Carlo Localization Using Adaptive Likelihood Models. In Proceedings of the 1st European Robotics Symposium (EUROS-06), Palermo, Italy, 16–18 March 2006; pp. 181–194.
6. Xia, L.L.; Cui, J.S.; Shen, R.; Xu, X.; Gao, Y.P.; Li, X.Y. A Survey of Image Semantics-based Visual Simultaneous Localization and Mapping: Application-oriented Solutions to Autonomous Navigation of Mobile Robots. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 4158. [CrossRef]
7. Tao, S.; Fengfeng, X.; Shuai, G.; Yu, L. Optimization of a Mobile Platform for a Wheeled Manipulator. *Robot. Auton. Syst.* **2016**, *8*, 061007. [CrossRef]
8. Stuckler, J.; Steffens, R.; Holz, D.; Behnke, S. Efficient 3D Object Perception and Grasp Planning for Mobile Manipulation in Domestic Environments. *Robot. Auton. Syst.* **2013**, *61*, 1106–1115. [CrossRef]
9. Meeussen, W.; Wise, M.; Glaser, S.; Chitta, S.; Mcgann, C.; Mihelich, P.; Mardereppstein, E.; Muja, M.; Eruhimov, V.; Foote, T.; et al. Autonomous Door Opening and Plugging in with a Personal Robot. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 729–736.
10. Welschehold, T.; Dornhege, C.; Burgard, W. Learning Mobile Manipulation Actions From Human Demonstrations. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 3196–3201.
11. Xiong, Y.; Ge, Y.Y.; Grimstad, L.; From, P.J. An Autonomous Atrawberry-harvesting Robot: Design, Development, Integration, and Field Evaluation. *J. Field Robot.*, **2020**, *37*, 202–224. [CrossRef]
12. Arad, B.; Balendonck, J.; Barth, R.; Ben-Shahar, O.; Edan, Y.; Hellstrom, T.; Hemming, J.; Kurtser, P.; Ringdahl, O.; Tielen, T.; et al. Development of a Sweet Pepper Harvesting Robot. *J. Field Robot.* **2020**, *37*, 1027–1039. [CrossRef]
13. Billard, A.; Kragic, D. Trends and Challenges in Robot Manipulation. *Science* **2019**, *364*, aat8414. [CrossRef] [PubMed]
14. Nieuwenhuisen, M.; Droeschel, D.; Holz, D.; Stuckler, J.; Berner, A.; Li, J.; Klein, R.; Behnke, S. Mobile Bin Picking with an Anthropomorphic Service Robot. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 2327–2334.
15. Asfour, T.; Wachter, M.; Kaul, L.; Rader, S.; Weiner, P.; Ottenhaus, S.; Grimm, R.; Zhou, Y.; Grotz, M.; Paus, F. ARMAR-6: A High-Performance Humanoid for Human-Robot Collaboration in Real-World Scenarios. *IEEE Robot. Autom. Mag.* **2019**, *26*, 108–121. [CrossRef]

16. You,Y.G.; Fan, Z.; Chen, W.Z.; Zhu, G.J.; Qiu, B.Z.; Xin, J.M.; Chen, J.M.; Deng, F.R.; Hou, Y.Z.; Liang, W.X.; et al. Design and Implementation of Mobile Manipulator System. In Proceedings of the 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Suzhou, China, 29 July–2 August 2019; pp. 113–118.
17. Klamt, T.; Schwarz, M.; Lenz, C.; Baccelliere, L.; Buongiorno, D.; Cichon, T.; DiGuardo, A.; Droeschel, D.; Gabardi, M.; Kamedula, M. Remote Mobile Manipulation with the Centauro Robot: Full-body Telepresence and Autonomous Operator Assistance. *J. Field Robot.* **2019**, *37*, 889–919. [CrossRef]
18. Klamt, T.; Rodriguez, D.; Baccelliere, L.; Chen, X.; Chiaradia, D.; Cichon, T.; Gabardi, M.; Guria, P.; Holmquist, K.; Kamedula, M.; et al. Flexible Disaster Response of Tomorrow: Final Presentation and Evaluation of the CENTAURO System. *IEEE Robot. Autom. Mag.* **2019**, *26*, 59–72. [CrossRef]
19. Orsag, M.; Korpela, C.; Oh, P. Modeling and Control of MM-UAV: Mobile Manipulating Unmanned Aerial Vehicle. *J. Intell. Robot. Syst.* **2013**, *29*, 227–240. [CrossRef]
20. Orsag, M.; Korpela, C.; Bogdan, S.; Oh, P. Dexterous Aerial Robots-Mobile Manipulation Using Unmanned Aerial Systems. *IEEE Trans. Robot.* **2017**, *33*, 1453–1466. [CrossRef]
21. Lindner, L.; Sergiyenko, O.; Rivas-Lopez, M.; Ivanov, M.; Mercorelli, P. Machine vision system errors for unmanned aerial vehicle navigation. In Proceedings of the 26th IEEE International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1615–1620.
22. Huang, H.; Tang, Q.R.; Li, J.Y.; Zhang, W.L.; Bao, X.; Zhu, H.T.; Wang, G. A Review on Underwater Autonomous Environmental Perception and Target Grasp, the Challenge of Robotic Organism Capture. *Ocean Eng.* **2020**, *195*, 106644. [CrossRef]
23. Eppner, C.; Hofer, S.; Jonschkowski, R.; Martin-Martin, R.; Sieverling, A.; Wall, V.; Brock, O. Four Aspects of Building Robotic Systems: Lessons From the Amazon Picking Challenge 2015. *Auton. Robot.* **2018**, *42*, 1459–1475. [CrossRef]
24. Correll, N.; Bekris, K.E.; Berenson, D.; Brock, O.; Causo, A.; Hauser, K.; Okada, K.; Rodriguez, A.; Romano, J.M.; Wurman, P.R. Analysis and Observations From the First Amazon Picking Challenge. *IEEE Trans. Autom. Sci. Eng.* **2018**, *1*, 172–188. [CrossRef]
25. Krotkov, E.; Hackett, D.; Jackel, L.; Perschbacher, M.; Pippine, J.; Strauss, J.; Pratt, G.; Orlowski, C. The DARPA Robotics Challenge Finals: Results and Perspectives. *J. Field Robot.* **2017**, *34*, 229–240. [CrossRef]
26. Han, Z.; Allspaw, J.; LeMasurier, G.; Parrillo, J.; Giger, D.; Ahmadzadeh, S.R.; Yanco, H.A. Towards Mobile Multi-Task Manipulation in a Confined and Integrated Environment with Irregular Objects. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–4 June 2020; pp. 11025–11031.
27. Yamamoto, T.; Terada, K.; Ochiai, A.; Saito, F.; Asahara, Y.; Murase, K. Development of the Research Platform of a Domestic Mobile Manipulator Utilized for International Competition and Field Test. In Proceedings of the 25th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 7675–7682.
28. Hamner, B.; Koterba, S.; Shi, J.; Simmons, R.; Singh, S. An autonomous mobile manipulator for assembly tasks. *Auton. Robot.* **2010**, *28*, 131–149. [CrossRef]
29. Chen, F.; Gao, B.Y.; Selvaggio, M.; Li, Z.J.; Caldwell, D.; Kershaw, K.; Masi, A.; Di Castro, M.; Losito, R. A Framework of Teleoperated and Stereo Vision Guided Mobile Manipulation for Industrial Automation. In Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation, Harbin, China, 7–10 August 2016; pp. 1641–1648.
30. Chen, F.; Selvaggio, M.; Caldwell, D.G. Dexterous Grasping by Manipulability Selection for Mobile Manipulator With Visual Guidance. *IEEE Trans. Ind. Inform.* **2019**, *15*, 1202–1210. [CrossRef]
31. Wang, C.; Zhang, Q.F.; Tian, Q.Y.; Li, S.; Wang, X.H.; Lane, D.; Petillot, Y.; Wang, S. Learning Mobile Manipulation Through Deep Reinforcement Learning. *Sensors* **2020**, *20*, 939. [CrossRef] [PubMed]
32. Saputra, M.R.U.; Markham, A.; Trigoni, N. Visual SLAM and Structure From Motion in Dynamic Environments: A Survey. *ACM Comput. Surv.* **2018**, *51*, 37. [CrossRef]
33. Sualeh, M.; Kim, G.W. Simultaneous Localization and Mapping in the Epoch of Semantics: A Survey. *Int. J. Control Autom. Syst.* **2019**, *17*, 729–742. [CrossRef]
34. Tzafestas, S.G. Mobile Robot Control and Navigation: A Global Overview. *J. Intell. Robot. Syst.* **2018**, *91*, 35–58. [CrossRef]

35. Kohlbrecher, S.; Meyer, J.; Graber, T.; Petersen, K.; Klingauf, U.; von Stryk, O. Hector Open Source Modules for Autonomous Mapping and Navigation with Rescue Robots. In Proceedings of the 17th International Symposium on Robot World Cup (RoboCup), Eindhoven, The Netherlands, 24 June–1 July 2013; pp. 624–631.
36. Kohlbrecher, S.; Von Stryk, O.; Meyer, J.; Klingauf, U. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In Proceedings of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–5 November 2011; pp. 155–160.
37. Wang, H.L.; Zhang, C.J.; Song, Y.; Pang, B.; Zhang, G.Y. Three-Dimensional Reconstruction Based on Visual SLAM of Mobile Robot in Search and Rescue Disaster Scenarios. *Robotica* **2020**, *38*, 350–373. [CrossRef]
38. Xu, X.S.; Liu, X.H.; Zhao, B.C.; Yang, B. An Extensible Positioning System for Locating Mobile Robots in Unfamiliar Environments. *Sensors* **2019**, *19*, 4025. [CrossRef]
39. Peng, J.C.; Shi, X.Y.; Wu, J.H.; Xiong, Z.H. An Object-Oriented Semantic SLAM System towards Dynamic Environments for Mobile Manipulation. In Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Hong Kong, 8–12 July 2019; pp. 199–204.
40. Zhang, J.; Shen, L. Clustering and Recognition for Automated Tracking and Grasping of Moving Objects. In Proceedings of the 2014 IEEE Workshop on Electronics, Computer and Applications (IWECA), Ottawa, ON, Canada, 8–9 May 2014; pp. 222–229.
41. Kim, S.; Shukla, A.; Billard, A. Catching Objects in Flight. *IEEE Trans. Robot.* **2014**, *30*, 1049–1065. [CrossRef]
42. Marturi, N.; Kopicki, M.; Rastegarpanah, A.; Rajasekaran, V.; Adjigble, M.; Stolkin, R.; Leonardis, A.; Bekiroglu, Y. Dynamic Grasp and Trajectory Planning for Moving Objects. *Auton. Robot.* **2019**, *43*, 1241–1256. [CrossRef]
43. Ardon, P.; Dragone, M.; Erden, M.S. Reaching and Grasping of Objects by Humanoid Robots through Visual Servoing. In Proceedings of the 11th International Conference on Haptics—Science, Technology, and Applications (EuroHaptics), Pisa, Italy, 13–16 June 2018; pp. 353–365.
44. Ma, Y.Q.; Liu, X.L.; Zhang, J.; Xu, D.; Zhang, D.P.; Wu, W.R. Robotic Grasping and Alignment for Small Size Components Assembly Based on Visual Servoing. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 4827–4843. [CrossRef]
45. De Miguel, M.A.; Garcia, F.; Armingol, J.M. Improved LiDAR Probabilistic Localization for Autonomous Vehicles Using GNSS. *Sensors* **2020**, *20*, 3145. [CrossRef] [PubMed]
46. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [CrossRef]
47. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
48. Zhang, H.; Tan, J.; Zhao, C.; Liang, Z.; Liu, L.; Zhong, H.; Fan, S. A Fast Detection and Grasping Method for Mobile Manipulator Based on Improved Faster R-CNN. *Ind. Robot.* **2020**, *47*, 167–175. [CrossRef]
49. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
50. Quigley, M.; Gerkey, B.; Conley, K.; Faust, J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the 2009 International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009. Available online: http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf (accessed on 19 November 2020).
51. Yang, T.J.; Howard, A.; Chen, B.; Zhang, X.; Go, A.; Sandler, M.; Sze, V.; Adam, H. NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications. In Proceedings of the 16th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 289–304.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.